

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
22 March 2001 (22.03.2001)

PCT

(10) International Publication Number
WO 01/20484 A2

(51) International Patent Classification⁷: G06F 17/00

(21) International Application Number: PCT/US00/25305

(22) International Filing Date:
15 September 2000 (15.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/153,960 15 September 1999 (15.09.1999) US

(71) Applicant: NETWORK SOLUTIONS, INC. [US/US];
505 Huntmar Park Drive, Herndon, VA 20170 (US).

(72) Inventors: CHAUHAN, Sanjeev; -. KORZENIEWSKI,
Greg; -. NAKHRE, Sujata; -. TAYLOR, Brian; -.
BREM, Larry; -. MOSTON, David; -.

(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Hen-
derson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street,
N.W., Washington, DC 20005-3315 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,
TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— Without international search report and to be republished
upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.

(54) Title: METHODS AND APPARATUS FOR ESTABLISHING AND MAINTAINING INTERNET DOMAIN NAME REGIS-
TRATIONS

(57) Abstract: Methods and apparatus for conducting transactions associated with managing Internet domain names includes es-
tablishing a connection between application programs of other users and a registrar of Internet domain names over the Internet,
authenticating the right of the user to so request fulfillment of a transaction, establishing a secure socket layer within the connection
and processing commands after such authentication.

WO 01/20484 A2

METHODS AND APPARATUS FOR ESTABLISHING
AND MAINTAINING INTERNET DOMAIN NAME
REGISTRATIONS

RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/153,960, filed September 15, 1999, the disclosure of which is expressly incorporated herein by reference. Copending U.S. Patent Application Serial No. 09/544,305, filed April 6, 2000, is also expressly
5 incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to the field of domain name registration. More particularly, the invention relates to methods and apparatus, which provides an application program interface to
10 enable the registration and maintenance of Internet domain names.

BACKGROUND OF THE INVENTION

The explosive growth of the Internet has created the need for orderly management of the process for creating domain name registrations (also known as Web addresses) and subsequent
15 administrative activities such as changes in ownership, renewals, and so forth. Until recently, Network Solutions, Inc. of Herndon, Virginia was the exclusive entity through which domain name registrations ("DNRs") were acquired and registered, at least for domain names ending in .com, .net, .org and .edu. In such role, Network Solutions, Inc. is known as a Registrar. Approximately five million Web addresses have been registered by Network Solutions, Inc. so far.

20 Presently-implemented systems for registration of Web Addresses rely almost exclusively on the use of e-mails between users and the Registrar for each step of the process. For example,

FIG. 4 shows the sequence of steps for registration, a customer (Registrant) prepares to register new Web Address 100;

The Registrant fills out a Service Agreement (102). The Service Agreement is submitted by e-mail to the Registrant, for example, to `hostmaster@internic.net` (step 104). The Request is automatically assigned a tracking number 106. The Service Agreement is automatically checked for errors 108. The Service Agreement is processed if there are no errors and the registrant is notified via e-mail when completed 112. Web Address registration is complete with a message. For example, the message may read: "Your Web Address registration is complete" (step 114). The customer is invoiced for the Web Address registration, step 116, and pays for the registration (step 118).

Finally, the Registrar sends a re-registration notice to registrant 60 days before the two year anniversary of the initial registration (step 120). Should there be an error in the Service Agreement, the registration process is not completed and an e-mail is sent to the customer informing it of the problem (step 110).

While this process works effectively, it is not without certain shortcomings. Most notably, the use of the Internet for e-mailing notices is not generally on a "real-time" basis. It may sometimes take several days for an e-mail to be delivered, particularly to parts of the world where the Internet support structure is not fully established. In addition, e-mails may become corrupted which creates further delays since the recipient must ask for the messages to be resent.

Further, recent events have resulted in the development of a concept termed "Wholesaling" by which other organizations, such as Internet Services Providers ("ISP"), Internet hosting providers, Internet portals and Electronic Commerce Providers are allowed to secure such DNRs on behalf of their own customers. In effect, such organizations act as their own Registrars by

offering their customers an interface which permits unsophisticated and sophisticated users to interface with the registration process and subsequent management thereof. Ultimately, however, DNRs are still issued by the Registrar, Network Solutions, Inc.

5 This expansion in the number of outlets for securing DNRs has led to the development of systems and methods to facilitate wholesaling.

Network Solutions, Inc. provides the NSI Wholesale Service Protocol ("WSP"), which establishes support of a real-time programmable interface for the purchase and management of Network Solutions, Inc. ("NSI") products and services a system. This system would permit users seeking registrations to complete the process, regardless of their level of expertise in computer
10 technology in general, and the Internet in particular. The WSP uses a simple, connection-based, synchronized, ASCII text-based protocol to communicate with an NSI Wholesale System over the Internet, thereby allowing Wholesalers to securely and reliably create, modify, and delete customer accounts and various products including the registration and management of Internet domain names.

15 The WSP consists of a set of commands that a Wholesaler may issue to create customers or domain-name-related products. Each command returns a set of 'properties' and major and minor error codes. In many cases, the result of one command is used as input to the next command.

Wholesale commands are either synchronous or asynchronous. Generally, customer creation and management commands are synchronous, while domain creation and management
20 commands are asynchronous. An "as-fast-as-possible" fulfillment system for asynchronous commands maintains queues in order to guarantee that the order is not lost in the case of external system problems. Typical synchronous response times are several seconds or less, while asynchronous times, for example, to complete a registration, are typically less than a few minutes.

There are two primary security mechanisms in the NSI Wholesale System. All Wholesalers must supply the IP addresses from which they will connect and establish Secure Socket Layer ("SSL") connections to the NSI Wholesale System. They also must use an account number and password when issuing session commands. NSI provides these ahead of time to Wholesalers. While this system provides numerous advantages over the prior art, there is a need to make the system universally available.

Accordingly, there is presently a need for methods and apparatus which provides an application program interface to allow computer application programs of other users, to access the Wholesale system and to conduct transactions associated with the registration and management of Internet domain names in the same manner as the WSP.

SUMMARY OF THE INVENTION

To achieve these objects and other advantages and in accordance with the purposes of the invention, as embodied and broadly described herein. The present invention is directed to methods and apparatus for interfacing a user application program to a system for conducting transactions associated with managing Internet domain names by establishing over the Internet a connection between the user and the system; authenticating the right of the user to conduct a transaction associated with managing Internet domain names; establishing a secure socket layer within the application program interface and the system; receiving a request in the application program interface from the applications program to initiate a transaction following such authentication; and transmitting commands by the application program interface to the system effectuating the transaction.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and together with the description, serve to explain the principles of the invention. In the drawings,

5 FIG. 1 is a diagram of a preferred embodiment of the invention, wherein an application program interface, WSP Java API, implements the functionality of a WSP in a series of Java object classes;

FIG. 2 shows an exemplary flowchart of the method for interfacing a user application program to a system for conducting transactions associated with managing Internet domain names;

10 FIG. 3 shows an exemplary diagram of the apparatus for interfacing a user application program to a system for conducting transactions associated with managing Internet domain names.

DETAILED DESCRIPTION

Reference will now be made in detail to the present preferred embodiment of the invention, an example of which is illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like elements.

The methods and apparatus provides an application program interface that is accessed by user application programs to establish a connection to a system for conducting transactions associated with managing Internet domain names.

20 Referring to FIG. 1, a computer application program 12 typically provides an interface to the user to collect customer data. The computer application program 12 typically issues a session request to the application program interface, WSP Java API 14, in order to establish a session over the Internet 10 with the Wholesale System 16 and to issue various commands.

The WSP Java API 14 is essentially a Java 'wrapper' of the WSP. It consists of two main classes: NSIWholesaleAPI and WSPResponse. The NSIWholesaleAPI contains all of the methods for creating customers, products, and services. WSPResponse is a single class returned from most methods and contains the result of the issued command request and will optionally include any additional data that was generated by the request.

One advantage of using WSP Java API 14 is that when the NSIWholesaleAPI object is created, the object creates a 'pool' of SSL connections to the Wholesale system 16. This class removes the need to issue a 'session' command, and removes the burden from a Wholesaler of making and managing SSL connections. Multi-threaded applications may also use this connection pool.

FIG. 2, shows an exemplary flowchart of the invention. A user 20, sends a request to the application program 12 to conduct a transaction associated with managing Internet domain names. The application program 12 typically provides an interface to a user to collect customer data. The application program 12 typically issues a session request to the application program interface 14 to establish a session over the Internet 10 with the Wholesale System 16. The application program interface 14 will authenticate a user's right to conduct a transaction by attempting to connect to the Wholesale System 16 using user account information received from the application program 12. If the authentication fails, the application program interface 14 will attempt two additional authentications. The application program interface 14 will send the result of the authentication to the application program 12. If the authentication is successful, the application program interface 14 will establish a secure socket layer with the Wholesale System 16. The application program 12 will send a request to the application program interface 14 to make available the set of commands through the instantiation of an object of class NSIWholesaleAPI. The application program 12

may now send a request to any of the object methods to request the processing of a command. The application program interface 14 will receive the command request along with any data. The application program interface 14 will format the data and the command into a string as described in detail in the aforementioned copending U.S. Patent Application Serial No. 09/544,305. The application program 14 will then transmit the request across the Internet 10 to the Wholesale System 16 to effectuate the transaction. The Wholesale System 16 will process the request and will return the result of the transaction across the Internet 10 to the application program interface 14. The application program interface 14 will send a response to the application program 12.

FIG. 3, is a diagram of an exemplary computer environment in which methods consistent with the present invention may be implemented. A component is provided for a user 20, to send a request to the application program 12 to conduct a transaction associated with managing Internet domain names. The application program 12 typically provides an interface to user 20 to collect customer data. The application program 12 typically issues a session request to the application program interface 14 to establish a session over the Internet 10 with the Wholesale System 16. A component is also provided for authenticating the right of the user 20 to conduct a transaction. The application program interface 14 will authenticate a user's right to conduct a transaction by attempting to connect to the Wholesale System 16 using the user account information received from the application program 12. If the authentication fails, the application program interface 14 will attempt two additional authentications. The application program interface 14 will send the result of the authentication to the application program 12. A component is also provided for establishing a secure socket layer if the authentication is successful. The application program interface 14 will establish a secure socket layer with the Wholesale System 16. A component is also provided for initiating a transaction following a successful authentication. The application

program 12 will send a request to the application program interface 14 to make available the set of commands that are available in the Wholesale System 16 through the instantiation of an object of class NSIWholesaleAPI. The application program 12 may now send a request to any of the object methods to request the processing of a command. The application program interface 14 will receive the command request along with any data. A component is also provided for transmitting commands through the secure socket layer. The application program interface 14 will format the data and command into a string as described in detail in the aforementioned copending U.S. Patent Application Serial No. 09/544,305. The application program 14 will then transmit the request across the Internet 10 to the Wholesale System 16 to effectuate the transaction. The Wholesale System 16 will process the request and return the result of the transaction across the Internet 10 to the application program interface 14. The application program interface 14 will send a response to the application program 12.

The name of the WSP Java API configuration file is typically WSPjavaAPI.conf. This file contains specific information about each Wholesale user. The values for the parameters in that file are provided when a Wholesale agreement is completed with a Registrar, such as NSI. The file includes the following parameters:

```
WholesaleAPI.minConnections=2
WholesaleAPI.maxConnections=2
WholesaleAPI.connectionSoTimeout=120000
WholesaleAPI.connectionProtocol=WSP1.5
WholesaleAPI.server=XXX.netsol.com
WholesaleAPI.port=X000
WholesaleAPI.retry=1

WholesaleAPI.WholesalerAccountId=12345678
WholesaleAPI.WholesalerPassword=XXXXXXXXX
```

The values indicated with an "X" will change from user to user. The definition of each of these parameters is as follows:

Configuration File Parameters

Name	Type	Typical Value	Definition
minConnections	Positive Integer	2 through 10	Connection Pooling Tuning Parameter
maxConnections	Positive Integer	minConnections through 5	Connection Pooling Tuning Parameter
connectionSoTimeout	Positive Integer	120000	Number of milliseconds of non-use before the API will time out a connection
Server	String	X.netsol.com	Server name for Wholesale protocol at Registrar. The exact name will be provided when the Wholesale agreement is established.
Port	Integer	XXXX	IP Port on the server for connection
WholesalerAccountId	String	XXX	Unique identifier of each Wholesale partner
WholesalerPassword	String	XXXXXXXX	Unique code, necessary for connection to NSI, for Wholesale operations

- o class java.lang.Object
 - o class nsi.wholesale.toolkit.javaapi.NSIWholesaleAPI
 - o class java.lang.Throwable (implements java.io.Serializable)
 - o class java.lang.Exception
 - o class nsi.wholesale.toolkit.javaapi.NSIWholesaleException
 - o class nsi.wholesale.toolkit.javaapi.WSPEException
 - o class nsi.wholesale.toolkit.javaapi.InvalidArgumentException
 - o class nsi.wholesale.toolkit.javaapi.InvalidResponseFormatException
 - o class nsi.wholesale.toolkit.javaapi.WSPResponse

The NSIWholesaleAPI class constructs an instance of a new NSIWholesaleAPI object whose methods can be called to perform Wholesale-related activities, such as connecting to the Wholesale System 16 over the Internet through a SSL, and it allows the computer application program 12 to create and manage Internet domain names. The class definition for NSIWholesaleAPI includes the following:

Constructor Summary	
<u>NSIWholesaleAPI</u> ()	
Method Summary	
<u>WSPResponse</u>	<u>createBusinessAccount</u> (<u>String</u> aCompanyName, <u>String</u> aCompanyType, <u>String</u> aAddress1, <u>String</u> aCity, <u>String</u> aCountryCode, <u>String</u> aPhone, <u>String</u> aCustomerPassword, <u>String</u> aAuthQuestion, <u>String</u> aAuthAnswer, <u>String</u> aLegalContactFirstName, <u>String</u> aLegalContactLastName) This method creates a business customer account and returns an automatically generated customer account id.
<u>WSPResponse</u>	<u>createBusinessAccount</u> (<u>String</u> aCompanyName, <u>String</u> aCompanyType, <u>String</u> aAddress1, <u>String</u> aCity, <u>String</u> aCountryCode, <u>String</u> aPhone, <u>String</u> aCustomerPassword,

	<u>String</u> aAuthQuestion, <u>String</u> aAuthAnswer, <u>String</u> aLegalContactFirstName, <u>String</u> aLegalContactLastName, <u>Properties</u> optionalAttributes) <p>This method creates a business customer account and returns an automatically generated customer account id.</p>
<u>WSPResponse</u>	<u>createBusinessAccount</u> (<u>String</u> aCompanyName, <u>String</u> aCompanyType, <u>String</u> aAddress1, <u>String</u> aCity, <u>String</u> aCountryCode, <u>String</u> aPhone, <u>String</u> aCustomerPassword, <u>String</u> aAuthQuestion, <u>String</u> aAuthAnswer, <u>String</u> aLegalContactFirstName, <u>String</u> aLegalContactLastName, <u>String</u> aState, <u>String</u> aPostalCode) <p>This method creates a business customer account and returns an automatically generated customer account id.</p>
<u>WSPResponse</u>	<u>createBusinessAccount</u> (<u>String</u> aCompanyName, <u>String</u> aCompanyType, <u>String</u> aAddress1, <u>String</u> aCity, <u>String</u> aCountryCode, <u>String</u> aPhone, <u>String</u> aCustomerPassword, <u>String</u> aAuthQuestion, <u>String</u> aAuthAnswer, <u>String</u> aLegalContactFirstName, <u>String</u> aLegalContactLastName, <u>String</u> aState, <u>String</u> aPostalCode, <u>Properties</u> optionalAttributes) <p>This method creates a business customer account and returns an automatically generated customer account id.</p>
<u>WSPResponse</u>	<u>createDnsHosting</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>String</u> aDomainName) <p>This method registers a DNS Hosting product for an existing customer account.</p>
<u>WSPResponse</u>	<u>createDnsHosting</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>String</u> aDomainName, <u>Properties</u> optionalAttributes) <p>This method registers a DNS Hosting product for an existing customer account.</p>
<u>WSPResponse</u>	<u>createDomain</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>String</u> aDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId) <p>This method registers a new domain product for an existing customer account.</p>
<u>WSPResponse</u>	<u>createDomain</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>String</u> aDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) <p>This method registers a new domain product for an existing customer account.</p>
<u>WSPResponse</u>	<u>createDomain</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>String</u> aDomainName, <u>String</u> aHostName1, <u>String</u> aHostAddr1, <u>String</u> aHostName2, <u>String</u> aHostAddr2, <u>String</u> aTechContactId) <p>This method registers a new domain product for an existing customer account.</p>

<u>WSPResponse</u>	<u>createDomain</u> (String pCustomerId, String pCustomerPassword, String aDomainName, String aHostName1, String aHostAddr1, String aHostName2, String aHostAddr2, String aTechContactId, Properties optionalAttributes) This method registers a new domain product for an existing customer account.
<u>WSPResponse</u>	<u>createIndividualAccount</u> (String aFirstName, String aLastName, String aAddress1, String aCountryCode, String aPhone, String aCustomerPassword, String aAuthQuestion, String aAuthAnswer, String aCity) This method creates an individual customer account and returns an automatically generated customer account id.
<u>WSPResponse</u>	<u>createIndividualAccount</u> (String aFirstName, String aLastName, String aAddress1, String aCountryCode, String aPhone, String aCustomerPassword, String aAuthQuestion, String aAuthAnswer, String aCity, Properties optionalAttributes) This method creates an individual customer account and returns an automatically generated customer account id.
<u>WSPResponse</u>	<u>createIndividualAccount</u> (String aFirstName, String aLastName, String aAddress1, String aCountryCode, String aPhone, String aCustomerPassword, String aAuthQuestion, String aAuthAnswer, String aCity, String aState, String aPostalCode) This method creates an individual customer account and returns an automatically generated customer account id.
<u>WSPResponse</u>	<u>createIndividualAccount</u> (String aFirstName, String aLastName, String aAddress1, String aCountryCode, String aPhone, String aCustomerPassword, String aAuthQuestion, String aAuthAnswer, String aCity, String aState, String aPostalCode, Properties optionalAttributes) This method creates an individual customer account and returns an automatically generated customer account id.
<u>WSPResponse</u>	<u>createTechnicalContact</u> (String aFirstName, String aLastName, String aOrganization, String aAddress1, String aCountryCode, String aPhone, String aEmail, String aTechContactPassword, String aCity) This method creates a technical contact and returns an automatically generated technical contact id (NIC Handle).
<u>WSPResponse</u>	<u>createTechnicalContact</u> (String aFirstName, String aLastName, String aOrganization, String aAddress1, String aCountryCode, String aPhone, String aEmail, String aTechContactPassword, String aCity, Properties optionalAttributes) This method creates a technical contact and returns an automatically generated technical contact id (NIC Handle).
<u>WSPResponse</u>	<u>createTechnicalContact</u> (String aFirstName, String aLastName, String aOrganization, String aAddress1, String aCountryCode, String aPhone, String aEmail, String aTechContactPassword,

	<u>String</u> aCity, <u>String</u> aState, <u>String</u> aPostalCode) This method creates a technical contact and returns an automatically generated technical contact id (NIC Handle).
<u>WSPResponse</u>	<u>createTechnicalContact</u> (<u>String</u> aFirstName, <u>String</u> aLastName, <u>String</u> aOrganization, <u>String</u> aAddress1, <u>String</u> aCountryCode, <u>String</u> aPhone, <u>String</u> aEmail, <u>String</u> aTechContactPassword, <u>String</u> aCity, <u>String</u> aState, <u>String</u> aPostalCode, <u>Properties</u> optionalAttributes) This method creates a technical contact and returns an automatically generated technical contact id (NIC Handle).
<u>WSPResponse</u>	<u>describe</u> (<u>String</u> pTarget) This method provides information about a specified target.
<u>WSPResponse</u>	<u>describe</u> (<u>String</u> pTarget, <u>Properties</u> optionalAttributes) This method provides information about a specified target.
<u>WSPResponse</u>	<u>generateDomainName</u> (<u>String</u> pKeyWords) This method is used to generate a list of available domain names based on a specified list of keywords.
<u>WSPResponse</u>	<u>generateDomainName</u> (<u>String</u> pKeyWords, <u>Properties</u> optionalAttributes) This method is used to generate a list of available domain names based on a specified list of keywords.
static <u>String</u>	<u>getJdkVersion</u> () Static accessor method to get JDK version
static long	<u>getSerialVersion</u> () Static accessor method to get serial version
static double	<u>getVersion</u> () Static accessor method to get release version
<u>WSPResponse</u>	<u>help</u> () This method returns basic information about the supported protocol commands and classes.
<u>WSPResponse</u>	<u>help</u> (<u>Properties</u> optionalAttributes) This method returns basic information about the supported protocol commands and classes.
static void	<u>init</u> (<u>Properties</u> properties) Initializes the Wholesale Services Protocol API.
static void	<u>init</u> (<u>String</u> configFileName) Initializes the Wholesale Services Protocol API.
<u>WSPResponse</u>	<u>lookupBusinessAccount</u> () Retrieves a list of all business customer accounts linked to wholesaler.
<u>WSPResponse</u>	<u>lookupBusinessAccount</u> (<u>Properties</u> optionalAttributes) Retrieves a list of all business customer accounts linked to wholesaler.

<u>WSPResponse</u>	<u>lookupBusinessAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword) Retrieves details of a specific business customer account.
<u>WSPResponse</u>	<u>lookupBusinessAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) Retrieves details of a specific business customer account.
<u>WSPResponse</u>	<u>lookupDnsHosting</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword) Retrieves a list of all domain names associated with DNS Hosting products owned by customer.
<u>WSPResponse</u>	<u>lookupDnsHosting</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) Retrieves a list of all domain names associated with DNS Hosting products owned by customer.
<u>WSPResponse</u>	<u>lookupDnsHosting</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName) Retrieves details of a specific DNS Hosting product owned by customer.
<u>WSPResponse</u>	<u>lookupDnsHosting</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>Properties</u> optionalAttributes) Retrieves details of a specific DNS Hosting product owned by customer.
<u>WSPResponse</u>	<u>lookupDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword) Retrieves a list of all domain products owned by customer.
<u>WSPResponse</u>	<u>lookupDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) Retrieves a list of all domain products owned by customer.
<u>WSPResponse</u>	<u>lookupDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName) Retrieves detail information about a specific domain product registered to a customer.
<u>WSPResponse</u>	<u>lookupDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>Properties</u> optionalAttributes) Retrieves detail information about a specific domain product registered to a customer.
<u>WSPResponse</u>	<u>lookupDomainByTechContact</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword) Retrieves a list of the customer's domain products for which the technical contact id is the technical contact.
<u>WSPResponse</u>	<u>lookupDomainByTechContact</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>Properties</u> optionalAttributes) Retrieves a list of the customer's domain products for which the

	technical contact id is the technical contact.
<u>WSPResponse</u>	<u>lookupDomainByTechContact</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName) Retrieves details of a specific domain product owned by customer, for which the technical contact id is the technical contact for the specific product.
<u>WSPResponse</u>	<u>lookupDomainByTechContact</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName, <u>Properties</u> optionalAttributes) Retrieves details of a specific domain product owned by customer, for which the technical contact id is the technical contact for the specific product.
<u>WSPResponse</u>	<u>lookupIndividualAccount</u> () Retrieves a list of all individual accounts linked to a wholesaler.
<u>WSPResponse</u>	<u>lookupIndividualAccount</u> (<u>Properties</u> optionalAttributes) Retrieves a list of all individual accounts linked to a wholesaler.
<u>WSPResponse</u>	<u>lookupIndividualAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword) Retrieves details of a specific individual customer account.
<u>WSPResponse</u>	<u>lookupIndividualAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) Retrieves details of a specific individual customer account.
<u>WSPResponse</u>	<u>lookupTechnicalContact</u> (<u>String</u> pTechContactId) Retrieves details of a specific technical contact.
<u>WSPResponse</u>	<u>lookupTechnicalContact</u> (<u>String</u> pTechContactId, <u>Properties</u> optionalAttributes) Retrieves details of a specific technical contact.
<u>WSPResponse</u>	<u>modifyBusinessAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword) This method is used to modify one or more attributes of an existing business (organization) customer account.
<u>WSPResponse</u>	<u>modifyBusinessAccount</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) This method is used to modify one or more attributes of an existing business (organization) customer account.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pIsppassword, <u>String</u> pDomainName, <u>String</u> aTechContactId) This method allows an Internet Service Provider (ISP) to modify the Technical Contact Id of a specific domain product.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pIsppassword, <u>String</u> pDomainName, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) This method allows an Internet Service Provider (ISP) to modify the Technical Contact Id of a specific domain product.

<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId) This method allows a customer to modify one or more attributes associated with a specific instance of a Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) This method allows a customer to modify one or more attributes associated with a specific instance of a Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId) This method allows a technical contact of a customer's domain product to modify one or more attributes associated with that specific instance of the Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName, <u>String</u> aHostId1, <u>String</u> aHostId2, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) This method allows a technical contact of a customer's domain product to modify one or more attributes associated with that specific instance of the Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>String</u> aHostName1, <u>String</u> aHostAddr1, <u>String</u> aHostName2, <u>String</u> aHostAddr2, <u>String</u> aTechContactId) This method allows a customer to modify one or more attributes associated with a specific instance of a Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pCustomerPassword, <u>String</u> pDomainName, <u>String</u> aHostName1, <u>String</u> aHostAddr1, <u>String</u> aHostName2, <u>String</u> aHostAddr2, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) This method allows a customer to modify one or more attributes associated with a specific instance of a Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName, <u>String</u> aHostName1, <u>String</u> aHostAddr1, <u>String</u> aHostName2, <u>String</u> aHostAddr2, <u>String</u> aTechContactId) This method allows a technical contact of a customer's domain product to modify one or more attributes associated with that specific instance of the Domain product registered to that customer.

<u>WSPResponse</u>	<u>modifyDomain</u> (<u>String</u> pCustomerAccountId, <u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>String</u> pDomainName, <u>String</u> aHostName1, <u>String</u> aHostAddr1, <u>String</u> aHostName2, <u>String</u> aHostAddr2, <u>String</u> aTechContactId, <u>Properties</u> optionalAttributes) This method allows a technical contact of a customer's domain product to modify one or more attributes associated with that specific instance of the Domain product registered to that customer.
<u>WSPResponse</u>	<u>modifyIndividualAccount</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword) This method is used to modify one or more attributes of an existing individual customer account.
<u>WSPResponse</u>	<u>modifyIndividualAccount</u> (<u>String</u> pCustomerAccountId, <u>String</u> pCustomerPassword, <u>Properties</u> optionalAttributes) This method is used to modify one or more attributes of an existing individual customer account.
<u>WSPResponse</u>	<u>modifyPassword</u> (<u>String</u> pCustomerAccountId, <u>String</u> aCustomerPassword) This method is used to modify/reset a customer's account password.
<u>WSPResponse</u>	<u>modifyPassword</u> (<u>String</u> pCustomerAccountId, <u>String</u> aCustomerPassword, <u>Properties</u> optionalAttributes) This method is used to modify/reset a customer's account password.
<u>WSPResponse</u>	<u>modifyTechnicalContact</u> (<u>String</u> pTechContactId, <u>String</u> pTechContactPassword) This method is used to modify one or more attributes of an existing technical contact.
<u>WSPResponse</u>	<u>modifyTechnicalContact</u> (<u>String</u> pTechContactId, <u>String</u> pTechContactPassword, <u>Properties</u> optionalAttributes) This method is used to modify one or more attributes of an existing technical contact.
<u>WSPResponse</u>	<u>quit</u> () This method terminates the wholesaler's access to session commands.
<u>WSPResponse</u>	<u>quit</u> (<u>Properties</u> optionalAttributes) This method terminates the wholesaler's access to session commands.
<u>void</u>	<u>shutdown</u> () This method shutdown the Wholesaler Services Protocol API.
<u>WSPResponse</u>	<u>verifyDomainName</u> (<u>String</u> pDomainName) This method verifies whether or not a specified domain name is available.
<u>WSPResponse</u>	<u>verifyDomainName</u> (<u>String</u> pDomainName, <u>Properties</u> optionalAttributes)

	This method verifies whether or not a specified domain name is available.
<u>WSPResponse</u>	<u>verifyDomains</u> (String pDomainTarget) This method verifies if series of domains are available.
<u>WSPResponse</u>	<u>verifyDomains</u> (String pDomainTarget, <u>Properties</u> optionalAttributes) This method verifies if series of domains are available.
<u>WSPResponse</u>	<u>verifyPassword</u> (String pCustomerAccountId, String pCustomerPassword) This method verifies whether or not a specified customer account id and password are valid.
<u>WSPResponse</u>	<u>verifyPassword</u> (String pCustomerAccountId, String pCustomerPassword, <u>Properties</u> optionalAttributes) This method verifies whether or not a specified customer account id and password are valid.

Methods inherited from class java.lang.Object
--

<u>equals</u> , <u>getClass</u> , <u>hashCode</u> , <u>notify</u> , <u>notifyAll</u> , <u>toString</u> , <u>wait</u> , <u>wait</u> , <u>wait</u>
--

5 The WSPResponse class represents the various components of a WSP response. A WSPResponse will be returned for command request that are issued through the application program interface 14. The response will describe, in as much detail as possible, the result of the issued request, and will optionally include any additional information that was generated by the request. A result consists of a "result code" followed by a descriptive "result message." The result code consists of a major result code followed by a period ('.'), followed by a minor result code.

The three-digit major result code provides a general indication of the success or failure of an issued request. In most cases, one or more result properties are returned. The three-digit minor result code may provide additional information about the major result code. Each command request has its own set of minor result code values or may have the value (000). The

15 WSPResponse and the result code are described in more detail in the aforementioned copending U.S. Patent Application Serial No. 09/544,305. A result property consists of a property key

followed by a colon (':'), followed by a property value. In some results, plain text is returned (e.g. information returned from the Help command). The result property is described in more detail in the aforementioned copending U.S. Patent Application Serial No. 09/544,305. The class definition for WSPResponse includes the following:

5

Constructor Summary	
<u>WSPResponse</u> (<u>String</u> data)	
Parses a WSP text response and extracts and stores the various components.	

Method Summary	
static void	<u>displayResponse</u> (<u>WSPResponse</u> response) prints debug-type representation of the WSP Response to standard output
static <u>String</u>	<u>getJdkVersion</u> () Static accessor method to get JDK version
int	<u>getMajorResultCode</u> () returns the WSP response major result code
int	<u>getMinorResultCode</u> () returns the WSP response minor result code
<u>Properties</u>	<u>getProperties</u> () Returns a property list containing all result property specifications, if applicable, extracted from the original WSP text response.
static <u>Properties</u>	<u>getProperties</u> (<u>WSPResponse</u> wspResponse) Returns a property list containing all result property specifications, if applicable, extracted from the original WSP text response.
<u>String</u>	<u>getProperty</u> (<u>String</u> resultPropertyKey) returns the result property value corresponding to the specified result property key, if found, extracted from the original WSP text response.
<u>String</u>	<u>getRawResponse</u> () Returns the original WSP text response string used to instantiate object.
<u>String</u>	<u>getResultCode</u> () returns the WSP response result code
<u>String</u>	<u>getResultCodeAndMessage</u> ()

	returns the WSP response result code and result message
<u>String</u>	<u>getResultMessage()</u> Returns the result message extracted from the original WSP text response.
static long	<u>getSerialVersion()</u> Static accessor method to get serial version
<u>Vector</u>	<u>getText()</u> returns a vector containing lines of text returned in the WSP response.
static double	<u>getVersion()</u> Static accessor method to get release version
<u>String</u>	<u>toString()</u> returns the raw WSP response

Methods inherited from class java.lang.Object
--

<u>equals</u> , <u>getClass</u> , <u>hashCode</u> , <u>notify</u> , <u>notifyAll</u> , <u>wait</u> , <u>wait</u> , <u>wait</u>
--

The NSIWholesaleException class is a user-defined exception class denoting a special error thrown by the methods in the class NSIWholesaleAPI. In particular, exceptions of the class NSIWholesaleException are thrown (a) when class initialization fails; (b) when a method in NSIWholesaleAPI is called without successfully initializing the class; (c) when a command cannot be completed (for example due to communications problems); or (d) to reframe an NSIException thrown at a lower level to give additional context information.

The class WSPException is the parent for wholesale protocol exception classes which processes responses sent by Wholesale Server 16 to computer application program 12. This is a service class, not used explicitly. It furnishes the base methods for storing error information common to all exception classes.

The classes InvalidArgumentException and InvalidResponseFormatException are the detail class children of WSPException. Both methods have the same internal functionality as

WSPException. In addition, both classes are independent, so that methods receiving the thrown exceptions know the type of error detected, even if the error context information is identical. In particular, the WSPResponse class throws IllegalArgumentException if the data returned by Wholesale Server 16 as part of a command, is invalid. The WSPResponse class throws class
 5 InvalidResponseFormatException if the data returned by the Wholesale Server 16 as part of a command, is not properly formatted. The class definition for NSIWholesaleException includes the following:

Constructor Summary	
<u>NSIWholesaleException</u> ()	
<u>NSIWholesaleException</u> (<u>String</u> message)	
<u>NSIWholesaleException</u> (<u>String</u> message, <u>Throwable</u> throwable)	
<u>NSIWholesaleException</u> (<u>Throwable</u> throwable)	

Method Summary	
static <u>String</u>	<u>getJdkVersion</u> () Static accessor method to get JDK version
static long	<u>getSerialVersion</u> () Static accessor method to get serial version
static double	<u>getVersion</u> () Static accessor method to get release version
<u>String</u>	<u>toString</u> ()

10

Methods inherited from class java.lang.Throwable	
<u>fillInStackTrace</u> , <u>getLocalizedMessage</u> , <u>getMessage</u> , <u>printStackTrace</u> , <u>printStackTrace</u> , <u>printStackTrace</u>	

Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, wait, wait, wait

The class definition for WSPException includes the following:

Constructor SummaryWSPException()WSPException(String message)WSPException(String message, Throwable x)

5

Method Summarystatic String getJdkVersion()

Static accessor method to get JDK version

String getMessage()static long getSerialVersion()

Static accessor method to get serial version

static double getVersion()

Static accessor method to get release version

Methods inherited from class java.lang.ThrowablefillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace, printStackTrace, toString**Methods inherited from class java.lang.Object**equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

10

The class definition for InvalidArgumentException is as follows:

Constructor SummaryInvalidArgumentException()

<u>InvalidArgumentException</u> (<u>String</u> message)
<u>InvalidArgumentException</u> (<u>String</u> message, <u>Throwable</u> x)

Method Summary	
static <u>String</u>	<u>getJdkVersion</u> () Static accessor method to get JDK version
<u>String</u>	<u>getMessage</u> ()
static long	<u>getSerialVersion</u> () Static accessor method to get serial version
static double	<u>getVersion</u> () Static accessor method to get release version

Methods inherited from class <u>java.lang.Throwable</u>
<u>fillInStackTrace</u> , <u>getLocalizedMessage</u> , <u>printStackTrace</u> , <u>printStackTrace</u> , <u>printStackTrace</u> , <u>toString</u>

Methods inherited from class <u>java.lang.Object</u>
<u>equals</u> , <u>getClass</u> , <u>hashCode</u> , <u>notify</u> , <u>notifyAll</u> , <u>wait</u> , <u>wait</u> , <u>wait</u>

5

The class definition for [InvalidResponseFormatException](#) includes the following:

Constructor Summary
<u>InvalidResponseFormatException</u> ()
<u>InvalidResponseFormatException</u> (<u>String</u> message)
<u>InvalidResponseFormatException</u> (<u>String</u> message, <u>Throwable</u> x)

Method Summary
static <u>String</u> <u>getJdkVersion</u> ()

	Static accessor method to get JDK version
<u>String</u>	<u>getMessage()</u>
static long	<u>getSerialVersion()</u> Static accessor method to get serial version
static double	<u>getVersion()</u> Static accessor method to get release version

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, printStackTrace, printStackTrace,
printStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

5

The invention will be further clarified by the following example showing a computer program file for creating a session with a Wholesale system to begin Wholesale operation, which is intended to be purely exemplary of the invention.

```

/**
 *----- Copyright (c) 2000 Network Solutions, Inc -----<BR>
 *
 * This class is a demonstration program for use case 1: Creating a session
 * with NSI to begin wholesale operations.
 *
 * This class assumes the existence of the file WSPjavaAPI.conf, which was set
 * up in running the installation test earlier.
 *
 *<BR>-----<BR>
 */
// Imports
import java.util.*;
import java.io.*;
import nsi.wholesale.toolkit.javaapi.util.resourcepool.*;
import nsi.wholesale.toolkit.javaapi.*;
//
class NSIUseCase1
{
//-----
public static void main(String args[])
{
    NSIWholesaleAPI api = null;
    // Use the properties from the configuration file to establish a session

```

```

try {
    NSIWholesaleAPI.init("WSPjavaAPI.conf");
}
catch (NSIWholesaleException e)
{
    System.out.print("Unable to initialize NSI Wholesale API:" + e);
    System.exit(1);
}
//Having initialized the NSI Wholesale API, we now need to
// create an instance of the API by calling constructor NSIWholesaleAPI()
try {
    api = new NSIWholesaleAPI();
}
catch (NSIWholesaleException e)
{
    System.out.print("Unable to initialize NSI Wholesale API:" + e);
    System.exit(1);
}
//
// At this point, if no error has been thrown, then the session
// is established, and a number of connections have been opened
// in the connection pool.
//
//
// Here, the call to methods on "api" are used to perform
// wholesale related activities. For this use case,
// there are none.
//
// End the session. This is done by calling the shutdown method on the
// instance of the wholesale API that was created earlier.
api.shutdown();
}
}

```

The WSP Java API 14 is an implementation of the WSP. In every case where the WSP Java API 14 is used, the first steps are to initialize a session and then create an instance of the class to reference the methods. Individual calls to methods are instances of a single protocol exchange.

Each call contains all necessary information (e.g., customer identification, contact information).

Except for session information, no information carries over from method to method.

As the example shows, the first evocation of a routine in the WSP Java API 14 is to the class method NSIWholesaleAPI.init to initialize the interface to the Wholesale System 16. The second evocation, creates an instance of the class NSIWholesaleAPI, which makes available to the application program 12 the methods to send commands to the Wholesale System 16. The application program 12 can now send command requests to the application program interface 14.

The last evocation is to the shutdown method, which ends the session with the Wholesale System
16.

It is to be noted that in the above description, references to Network Solutions, Inc. and
NSI, including but not limited to references which are imbedded in the examples given of classes,
5 methods, constructors, drawings, and the like, are intended to be exemplary only and not restricted
to any particular implementation.

While the foregoing description is based on a client-server architecture, those skilled in the
art will recognize that other types of architecture may be used consistent with the invention. In
10 addition, while the foregoing description is based on a software implementation, other
implementations may be used consistent with the invention. Further, while the foregoing
description is based on the Java language, those skilled in the art will recognize that other types of
language may be used consistent with the invention. Therefore, it is intended that this invention
not be limited to the particular implementation and method disclosed herein, but that the invention
15 include all implementations falling within the scope of the appended claims.

The foregoing description is presented for illustration and explanation. It is not intended to
be exhaustive or to limit the invention to the precise form disclosed, and modifications of
variations are possible in light of the above teachings or may be acquired from practice of the
invention. The principles of the invention and its practical application enable one skilled in the art
20 to utilize the invention in various embodiments and with various modifications as are suited to the
use contemplated.

The class definition for NSIWholesaleAPI is as follows:

```
//-----  
// Package  
25 //-----  
package nsi.wholesale.toolkit.javaapi;
```



```

//-----
// Imports...
//-----
5  import java.io.LineNumberReader;
   import java.io.FileReader;
   import java.io.IOException;
   import java.io.FileNotFoundException;
10  import java.util.StringTokenizer;
   import java.util.Properties;
   import java.util.Enumeration;
   import java.util.NoSuchElementException;
   import nsi.wholesale.toolkit.javaapi.util.resourcepool.Resource;
15  import nsi.wholesale.toolkit.javaapi.util.resourcepool.ResourceException;

/**
 *----- Copyright (c) 2000 Network Solutions, Inc -----
 *-----<BR>
20  *
 * This Java-based Wholesale Services Protocol (WSP) API provides
 * programmer access to the Wholesale Services Protocol Server.
 * The Wholesale Services Protocol (WSP) is a simple, connection-based,
 * ASCII text-based protocol. It allows wholesalers to securely and
25  * reliably create, modify, and lookup customer accounts and various
 * products via a single client/server interface.
 *
 *
 *<BR>-----
30  <BR>
 * $Log$
 *<BR>-----
<BR>
 */
35  public class NSIWholesaleAPI
   {
   //-----
   // Beginning of Required/Standard Fields
   //-----
40      /** Serial Version */
      static long serialVersion = 1L;

      /** Release Version */
      static double version = 1.1D;
45
      /** SCCS control ID */
      static final String sccsID = "@(#) wholesale-sdk_d1.12.1.3 08/04/00
10:27:19 NSIWholesaleAPI.java NSI";

50      /** JDK Version */
      static String jdkVersion = "1.1.7";

      // Static Accessor Method(s)...
      /** Static accessor method to get serial version */
55      static public long getSerialVersion() { return serialVersion; }

      /** Static accessor method to get release version */

```

```

static public double getVersion() { return version; }

    /** Static accessor method to get JDK version */
    static public String getJdkVersion() { return jdkVersion; }
5  //-----
    // End of Required/Standard Fields
    //-----

10 //-----
    // class (static) variable(s) ...
    //-----
        /** The name of this class */
        private static final String className = "NSIWholesaleAPI";

15        /** Connection pool */
        protected static NSIWholesalePool pool = null;

        /** Retry count */
        protected static int retry = 1;

20        // Supported Commands
        //-----
        /** Ease-of-use constant for Command: Session */
        public static final String COMMAND_SESSION = "Session";
25        /** Ease-of-use constant for Command: Describe */
        public static final String COMMAND_DESCRIBE = "Describe";
        /** Ease-of-use constant for Command: Verify */
        public static final String COMMAND_VERIFY = "Verify";
        /** Ease-of-use constant for Command: Create */
30        public static final String COMMAND_CREATE = "Create";
        /** Ease-of-use constant for Command: Delete */
        public static final String COMMAND_DELETE = "Delete";
        /** Ease-of-use constant for Command: Modify */
        public static final String COMMAND_MODIFY = "Modify";
35        /** Ease-of-use constant for Command: Renew */
        public static final String COMMAND_RENEW = "Renew";
        /** Ease-of-use constant for Command: Generate */
        public static final String COMMAND_GENERATE = "Generate";
        /** Ease-of-use constant for Command: Lookup */
40        public static final String COMMAND_LOOKUP = "Lookup";
        /** Ease-of-use constant for Command: Help */
        public static final String COMMAND_HELP = "Help";
        /** Ease-of-use constant for Command: Quit */
        public static final String COMMAND_QUIT = "Quit";

45        // Supported Classes
        //-----
        /** Ease-of-use constant for Class: DomainName */
        public static final String CLASS_DOMAIN_NAME = "DomainName";
50        /** Ease-of-use constant for Class: Domains */
        public static final String CLASS_DOMAINS = "Domains";
        /** Ease-of-use constant for Class: Password */
        public static final String CLASS_PASSWORD = "Password";
        /** Ease-of-use constant for Class: IndividualAccount */
55        public static final String CLASS_INDIVIDUAL_ACCOUNT =
"IndividualAccount";
        /** Ease-of-use constant for Class: BusinessAccount */

```

```

    public static final String CLASS_BUSINESS_ACCOUNT = "BusinessAccount";
    /** Ease-of-use constant for Class: TechnicalContact */
    public static final String CLASS_TECHNICAL_CONTACT = "TechnicalContact";
    /** Ease-of-use constant for Class: Domain */
5    public static final String CLASS_DOMAIN = "Domain";
    /** Ease-of-use constant for Class: DnsHosting */
    public static final String CLASS_DNS_HOSTING = "DnsHosting";

    // Supported Command-Class Combinations
10    //-----
    /** Ease-of-use constant for Command-Class Combination: VerifyDomainName
    */
    public static final String COMBO_VERIFY_DOMAIN_NAME = COMMAND_VERIFY + "
" + CLASS_DOMAIN_NAME;
15    /** Ease-of-use constant for Command-Class Combination: Verify Domains
    */
    public static final String COMBO_VERIFY_DOMAINS = COMMAND_VERIFY + " " +
CLASS_DOMAINS;
    /** Ease-of-use constant for Command-Class Combination: Verify Password
20    */
    public static final String COMBO_VERIFY_PASSWORD = COMMAND_VERIFY + " "
+ CLASS_PASSWORD;

    /** Ease-of-use constant for Command-Class Combination: Create
25    IndividualAccount */
    public static final String COMBO_CREATE_INDIVIDUAL_ACCOUNT =
COMMAND_CREATE + " " + CLASS_INDIVIDUAL_ACCOUNT;
    /** Ease-of-use constant for Command-Class Combination: Create
BusinessAccount */
30    public static final String COMBO_CREATE_BUSINESS_ACCOUNT =
COMMAND_CREATE + " " + CLASS_BUSINESS_ACCOUNT;
    /** Ease-of-use constant for Command-Class Combination: Create
TechnicalContact */
    public static final String COMBO_CREATE_TECHNICAL_CONTACT =
35    COMMAND_CREATE + " " + CLASS_TECHNICAL_CONTACT;
    /** Ease-of-use constant for Command-Class Combination: Create Domain */
    public static final String COMBO_CREATE_DOMAIN = COMMAND_CREATE + " " +
CLASS_DOMAIN;
    /** Ease-of-use constant for Command-Class Combination: Create
40    DnsHosting */
    public static final String COMBO_CREATE_DNS_HOSTING = COMMAND_CREATE + "
" + CLASS_DNS_HOSTING;

    /** Ease-of-use constant for Command-Class Combination: Delete Domain */
45    public static final String COMBO_DELETE_DOMAIN = COMMAND_DELETE + " " +
CLASS_DOMAIN;
    /** Ease-of-use constant for Command-Class Combination: Delete
DnsHosting */
    public static final String COMBO_DELETE_DNS_HOSTING = COMMAND_DELETE + "
50    " + CLASS_DNS_HOSTING;
    /** Ease-of-use constant for Command-Class Combination: Delete
IndividualAccount */
    public static final String COMBO_DELETE_INDIVIDUAL_ACCOUNT =
COMMAND_DELETE + " " + CLASS_INDIVIDUAL_ACCOUNT;
55    /** Ease-of-use constant for Command-Class Combination: Delete
BusinessAccount */

```

```

        public static final String COMBO_DELETE_BUSINESS_ACCOUNT =
COMMAND_DELETE + " " + CLASS_BUSINESS_ACCOUNT;

        /** Ease-of-use constant for Command-Class Combination: Modify Domain */
5        public static final String COMBO_MODIFY_DOMAIN = COMMAND_MODIFY + " " +
CLASS_DOMAIN;
        /** Ease-of-use constant for Command-Class Combination: Modify
IndividualAccount */
        public static final String COMBO_MODIFY_INDIVIDUAL_ACCOUNT =
10        COMMAND_MODIFY + " " + CLASS_INDIVIDUAL_ACCOUNT;
        /** Ease-of-use constant for Command-Class Combination: Modify
BusinessAccount */
        public static final String COMBO_MODIFY_BUSINESS_ACCOUNT =
COMMAND_MODIFY + " " + CLASS_BUSINESS_ACCOUNT;
15        /** Ease-of-use constant for Command-Class Combination: Modify
TechnicalContact */
        public static final String COMBO_MODIFY_TECHNICAL_CONTACT =
COMMAND_MODIFY + " " + CLASS_TECHNICAL_CONTACT;
        /** Ease-of-use constant for Command-Class Combination: Modify Password
20        */
        public static final String COMBO_MODIFY_PASSWORD = COMMAND_MODIFY + " "
+ CLASS_PASSWORD;

        /** Ease-of-use constant for Command-Class Combination: Renew Domain */
25        public static final String COMBO_RENEW_DOMAIN = COMMAND_RENEW + " " +
CLASS_DOMAIN;
        /** Ease-of-use constant for Command-Class Combination: Renew DnsHosting
*/
        public static final String COMBO_RENEW_DNS_HOSTING = COMMAND_RENEW + " "
30        + CLASS_DNS_HOSTING;

        /** Ease-of-use constant for Command-Class Combination: Generate
DomainName */
        public static final String COMBO_GENERATE_DOMAIN_NAME = COMMAND_GENERATE
35        + " " + CLASS_DOMAIN_NAME;

        /** Ease-of-use constant for Command-Class Combination: Lookup Domain */
        public static final String COMBO_LOOKUP_DOMAIN = COMMAND_LOOKUP + " " +
CLASS_DOMAIN;
40        /** Ease-of-use constant for Command-Class Combination: Lookup
DnsHosting */
        public static final String COMBO_LOOKUP_DNS_HOSTING = COMMAND_LOOKUP + "
" + CLASS_DNS_HOSTING;
        /** Ease-of-use constant for Command-Class Combination: Lookup
45        IndividualAccount */
        public static final String COMBO_LOOKUP_INDIVIDUAL_ACCOUNT =
COMMAND_LOOKUP + " " + CLASS_INDIVIDUAL_ACCOUNT;
        /** Ease-of-use constant for Command-Class Combination: Lookup
BusinessAccount */
50        public static final String COMBO_LOOKUP_BUSINESS_ACCOUNT =
COMMAND_LOOKUP + " " + CLASS_BUSINESS_ACCOUNT;
        /** Ease-of-use constant for Command-Class Combination: Lookup
TechnicalContact */
        public static final String COMBO_LOOKUP_TECHNICAL_CONTACT =
55        COMMAND_LOOKUP + " " + CLASS_TECHNICAL_CONTACT;

        // Supported Attributes

```

```

//-----
/** Ease-of-use constant for Attribute Key: FirstName */
public static final String ATTR_FIRST_NAME = "FirstName";
/** Ease-of-use constant for Attribute Key: LastName */
5 public static final String ATTR_LAST_NAME = "LastName";
/** Ease-of-use constant for Attribute Key: Address1 */
public static final String ATTR_ADDRESS1 = "Address1";
/** Ease-of-use constant for Attribute Key: Address2 */
public static final String ATTR_ADDRESS2 = "Address2";
10 /** Ease-of-use constant for Attribute Key: Address3 */
public static final String ATTR_ADDRESS3 = "Address3";
/** Ease-of-use constant for Attribute Key: Address4 */
public static final String ATTR_ADDRESS4 = "Address4";
/** Ease-of-use constant for Attribute Key: Address5 */
15 public static final String ATTR_ADDRESS5 = "Address5";
/** Ease-of-use constant for Attribute Key: City */
public static final String ATTR_CITY = "City";
/** Ease-of-use constant for Attribute Key: State */
public static final String ATTR_STATE = "State";
20 /** Ease-of-use constant for Attribute Key: PostalCode */
public static final String ATTR_POSTAL_CODE = "PostalCode";
/** Ease-of-use constant for Attribute Key: CountryCode */
public static final String ATTR_COUNTRY_CODE = "CountryCode";
/** Ease-of-use constant for Attribute Key: Phone */
25 public static final String ATTR_PHONE = "Phone";
/** Ease-of-use constant for Attribute Key: Fax */
public static final String ATTR_FAX = "Fax";
/** Ease-of-use constant for Attribute Key: Email */
public static final String ATTR_EMAIL = "Email";
30 /** Ease-of-use constant for Attribute Key: CustomerAccountId */
public static final String ATTR_CUSTOMER_ACCOUNT_ID =
"CustomerAccountId";
/** Ease-of-use constant for Attribute Key: CustomerPassword */
public static final String ATTR_CUSTOMER_PASSWORD = "CustomerPassword";
35 /** Ease-of-use constant for Attribute Key: AuthQuestion */
public static final String ATTR_AUTH_QUESTION = "AuthQuestion";
/** Ease-of-use constant for Attribute Key: AuthAnswer */
public static final String ATTR_AUTH_ANSWER = "AuthAnswer";
/** Ease-of-use constant for Attribute Key: CompanyName */
40 public static final String ATTR_COMPANY_NAME = "CompanyName";
/** Ease-of-use constant for Attribute Key: CompanyType */
public static final String ATTR_COMPANY_TYPE = "CompanyType";
/** Ease-of-use constant for Attribute Key: LegalContactFirstName */
public static final String ATTR_LEGAL_CONTACT_FIRST_NAME =
45 "LegalContactFirstName";
/** Ease-of-use constant for Attribute Key: LegalContactLastName */
public static final String ATTR_LEGAL_CONTACT_LAST_NAME =
"LegalContactLastName";
/** Ease-of-use constant for Attribute Key: LegalContactAddress1 */
50 public static final String ATTR_LEGAL_CONTACT_ADDRESS1 =
"LegalContactAddress1";
/** Ease-of-use constant for Attribute Key: LegalContactAddress2 */
public static final String ATTR_LEGAL_CONTACT_ADDRESS2 =
"LegalContactAddress2";
55 /** Ease-of-use constant for Attribute Key: LegalContactAddress3 */
public static final String ATTR_LEGAL_CONTACT_ADDRESS3 =
"LegalContactAddress3";

```

```

    /** Ease-of-use constant for Attribute Key: LegalContactAddress4 */
    public static final String ATTR_LEGAL_CONTACT_ADDRESS4 =
"LegalContactAddress4";
    /** Ease-of-use constant for Attribute Key: LegalContactAddress5 */
5    public static final String ATTR_LEGAL_CONTACT_ADDRESS5 =
"LegalContactAddress5";
    /** Ease-of-use constant for Attribute Key: LegalContactCity */
    public static final String ATTR_LEGAL_CONTACT_CITY = "LegalContactCity";
    /** Ease-of-use constant for Attribute Key: LegalContactState */
10    public static final String ATTR_LEGAL_CONTACT_STATE =
"LegalContactState";
    /** Ease-of-use constant for Attribute Key: LegalContactPostalCode */
    public static final String ATTR_LEGAL_CONTACT_POSTAL_CODE =
"LegalContactPostalCode";
15    /** Ease-of-use constant for Attribute Key: LegalContactCountryCode */
    public static final String ATTR_LEGAL_CONTACT_COUNTRY_CODE =
"LegalContactCountryCode";
    /** Ease-of-use constant for Attribute Key: LegalContactPhone */
    public static final String ATTR_LEGAL_CONTACT_PHONE =
20    "LegalContactPhone";
    /** Ease-of-use constant for Attribute Key: LegalContactFax */
    public static final String ATTR_LEGAL_CONTACT_FAX = "LegalContactFax";
    /** Ease-of-use constant for Attribute Key: LegalContactEmail */
    public static final String ATTR_LEGAL_CONTACT_EMAIL =
25    "LegalContactEmail";
    /** Ease-of-use constant for Attribute Key: Organization */
    public static final String ATTR_ORGANIZATION = "Organization";
    /** Ease-of-use constant for Attribute Key: TechContactPassword */
    public static final String ATTR_TECH_CONTACT_PASSWORD =
30    "TechContactPassword";
    /** Ease-of-use constant for Attribute Key: DomainName */
    public static final String ATTR_DOMAIN_NAME = "DomainName";
    /** Ease-of-use constant for Attribute Key: HostName1 */
    public static final String ATTR_HOST_NAME1 = "HostName1";
35    /** Ease-of-use constant for Attribute Key: HostAddr1 */
    public static final String ATTR_HOST_ADDR1 = "HostAddr1";
    /** Ease-of-use constant for Attribute Key: HostName2 */
    public static final String ATTR_HOST_NAME2 = "HostName2";
    /** Ease-of-use constant for Attribute Key: HostAddr2 */
40    public static final String ATTR_HOST_ADDR2 = "HostAddr2";
    /** Ease-of-use constant for Attribute Key: HostId1 */
    public static final String ATTR_HOST_ID1 = "HostId1";
    /** Ease-of-use constant for Attribute Key: HostId2 */
    public static final String ATTR_HOST_ID2 = "HostId2";
45    /** Ease-of-use constant for Attribute Key: TechContactId */
    public static final String ATTR_TECH_CONTACT_ID = "TechContactId";
    /** Ease-of-use constant for Attribute Key: IspPassword */
    public static final String ATTR_ISP_PASSWORD = "IspPassword";
    /** Ease-of-use constant for Attribute Key: Protocol (returned by
50 Describe) */
    public static final String ATTR_PROTOCOL = "Protocol";
    /** Ease-of-use constant for Attribute Key: Dns Hosting Created
    (returned by Create DnsHosting) */
    public static final String ATTR_DNS_HOSTING_CREATED = "Dns Hosting
55 Created";

    // Supported Parameters

```

```

//-----
/** Ease-of-use constant for Parameter Key: -WholesalerAccountId */
public static final String PARM_WHOLESALE_ACCOUNT_ID = "-
WholesalerAccountId";
5      /** Ease-of-use constant for Parameter Key: -WholesalerPassword */
public static final String PARM_WHOLESALE_PASSWORD = "-
WholesalerPassword";
      /** Ease-of-use constant for Parameter Key: -Target */
public static final String PARM_TARGET = "-Target";
10     /** Ease-of-use constant for Parameter Key: -DomainTarget */
public static final String PARM_DOMAIN_TARGET = "-DomainTarget";
      /** Ease-of-use constant for Parameter Key: -KeyWords */
public static final String PARM_KEY_WORDS = "-KeyWords";
      /** Ease-of-use constant for Parameter Key: -CustomerAccountId */
15     public static final String PARM_CUSTOMER_ACCOUNT_ID = "-" +
ATTR_CUSTOMER_ACCOUNT_ID;
      /** Ease-of-use constant for Parameter Key: -CustomerPassword */
public static final String PARM_CUSTOMER_PASSWORD = "-" +
ATTR_CUSTOMER_PASSWORD;
20     /** Ease-of-use constant for Parameter Key: -DomainName */
public static final String PARM_DOMAIN_NAME = "-" + ATTR_DOMAIN_NAME;
      /** Ease-of-use constant for Parameter Key: -TechContactId */
public static final String PARM_TECH_CONTACT_ID = "-" +
ATTR_TECH_CONTACT_ID;
25     /** Ease-of-use constant for Parameter Key: -TechContactPassword */
public static final String PARM_TECH_CONTACT_PASSWORD = "-" +
ATTR_TECH_CONTACT_PASSWORD;
      /** Ease-of-use constant for Parameter Key: -IspPassword */
public static final String PARM_ISP_PASSWORD = "-" + ATTR_ISP_PASSWORD;
30

//-----
// class (static) method(s) ...
//-----
35     /**
     * Initializes the Wholesale Services Protocol API.
     *
     * @param String name of configuration files containing property list
     * @return void
40     * @see NSIWholesaleAPI#init
     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException if
     initialization fails
     */
public static void init(String configFileName)
45     throws NSIWholesaleException {
    System.out.println("NSIWholesaleAPI.init(String) started");
    Properties configProps = new Properties();
    try {
50         LineNumberReader in = new LineNumberReader(new
        FileReader(configFileName));
        String line = in.readLine();
        while (line != null) {
            line = line.trim();
            if ( (line.length() > 0) && (line.startsWith("#") ==
55         false) ) {
                try {

```

```

StringTokenizer st = new
StringTokenizer(line, "=");
String propertyKey = st.nextToken();
String propertyValue = st.nextToken();
configProps.put(propertyKey,
5   propertyValue);
    }
    catch (NoSuchElementException nsee) {
10       System.err.println("NSIWholesaleAPI.init(String): Error Reading Config
File" + nsee);
        throw new NSIWholesaleException("Error
reading Configuration File " + configFileName);
    }
15   }

    line = in.readLine();
    }
    }
20   catch (FileNotFoundException fnfe) {
        System.err.println("NSIWholesaleAPI.init(String):
Configuration File Not Found " + fnfe);
        throw new NSIWholesaleException("Configuration File Not
Found " + configFileName);
25   }
    catch (IOException ioe) {
        System.err.println("NSIWholesaleAPI.init(String): IO
Exception reading configuration file " + ioe);
        throw new NSIWholesaleException("IO Exception reading
30   configuration file " + configFileName);
    }
    catch (Exception e) {
        System.err.println("NSIWholesaleAPI.init(String): Exception
reading configuration file " + e);
35   throw new NSIWholesaleException("Exception reading
configuration file " + configFileName);
    }

    NSIWholesaleAPI.init(configProps);
40   System.out.println("NSIWholesaleAPI.init(String) completed
successfully");
}

45   /**
    * Initializes the Wholesale Services Protocol API.
    * <BR>
    * All of the following parameters should be
    * passed in through the properties file:
50   <PRE>
        WholesaleAPI.minConnections -
            Connections to use on startup (1,2,5, 10)
        WholesaleAPI.maxConnections -
            Maximum Connections (5,10,20)
55   WholesaleAPI.connectionSoTimeout - (Provided by NSI)
            Enable/disable Socket SO_TIMEOUT with the specified
timeout, in milliseconds

```



```

        WholesaleAPI.connectionProtocol - (Provided by NSI)
        Wholesale Services Protocol Version
        WholesaleAPI.server - (Provided by NSI)
        The secure wholesale server at NSI
5      WholesaleAPI.port - (Provided by NSI)
        The port on that machine
        WholesaleAPI.retry -
        The retry count, if a transient error occurs (2,3)

10     WholesaleAPI.WholesalerAccountId - (Provided by NSI)
        The Wholesaler Account number
        WholesaleAPI.WholesalerPassword - (Provided by NSI)
        The Password for that account

        </PRE>
15     *
    * @param properties Properties file containing configuration settings
    * @return void
    * @see NSIWholesaleAPI#NSIWholesaleAPI
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException if
20 initialization fails
    */
    public static void init(Properties properties)
        throws NSIWholesaleException {
        System.out.println("NSIWholesaleAPI.init(Properties) started");
25     int min = parseInt(
        properties.getProperty("WholesaleAPI.minConnections"));
        int max = parseInt(
        properties.getProperty("WholesaleAPI.maxConnections"));
        int connectionSoTimeout = parseInt(
30     properties.getProperty("WholesaleAPI.connectionSoTimeout"));
        String connectionProtocol =
        properties.getProperty("WholesaleAPI.connectionProtocol");
        String server = properties.getProperty("WholesaleAPI.server");
        int port = parseInt(properties.getProperty("WholesaleAPI.port"));
35     retry = parseInt(properties.getProperty("WholesaleAPI.retry"));

        String wholesalerAccountId =
        properties.getProperty("WholesaleAPI.WholesalerAccountId");
        String wholesalerPassword =
40     properties.getProperty("WholesaleAPI.WholesalerPassword");

        System.out.println("connectionProtocol = " + connectionProtocol);
        System.out.println("server = " + server);
45     System.out.println("port = " + port);
        System.out.println("wholesalerAccountId = " +
        wholesalerAccountId);
        System.out.println("wholesalerPassword = " + wholesalerPassword);
        System.out.println("NSIWholesaleAPI.init(Properties) constructing
50 new NSIWholesalePool");
        pool = new NSIWholesalePool(server, port, min, max,
        wholesalerAccountId, wholesalerPassword, retry,
        connectionSoTimeout);
        NSIWholesaleAPI api = new NSIWholesaleAPI();
55     if (api.verifyConnectionProtocol(connectionProtocol) != true)
        {
            api = null;

```

```

        throw new NSIWholesaleException("Failed to verify connection
protocol " + connectionProtocol);
    }
    api = null;
5      System.out.println("NSIWholesaleAPI.init(Properties) completed
successfully");
    }

10     //-----
    // static initializer block ...
    //-----

15     //-----
    // instance Variable(s) ...
    //-----

20     //-----
    // Constructor(s)
    //-----
    /**
     * Constructs a new NSIWholesaleAPI object. NSIWholesaleAPI MUST have
     * been initialized BEFORE any NSIWholesaleAPI objects can be
constructed.
25     * @see NSIWholesaleAPI#init
     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException if
NSIWholesaleAPI has not yet
     * been initialized
    */
30     public NSIWholesaleAPI() throws NSIWholesaleException {
        if (pool == null)
            throw new NSIWholesaleException("Must initialize API before
use");
    }

35     //-----
    // Accessors and Mutators
    //-----

40     //-----
    // General Methods
    //-----
    //##--BEGIN API-----
45     -----
    /**
     * This method is used to authenticate a wholesaler and to provide
     * the wholesaler with access to Session commands. Once a session
     * has been established, the Session command may not be issued
50     * again
     * <BR><PRE>
        Expected Major Result Code: SESSION_CREATED
     * </PRE><BR>
     * @param pWholesalerAccountId the parameter value
55     * @param pWholesalerPassword the parameter value
     * @return the Wholesale Server Protocol Server response

```

```

        * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
        * @see COMMAND_SESSION
        * @see PARM_WHOLESALE_ACCOUNT_ID
5        * @see PARM_WHOLESALE_PASSWORD
        * @see WSPResponse#SESSION_CREATED
        * @see WSPResponse#AUTHENTICATION_FAILED
        *
    */
10    WSPResponse session(
        String pWholesalerAccountId,
        String pWholesalerPassword)
        throws NSIWholesaleException {
        return session( pWholesalerAccountId, pWholesalerPassword, new
15    Properties(), 0);
    }
    /**
        * This method is used to authenticate a wholesaler and to provide
        * the wholesaler with access to Session commands. Once a session
20        * has been established, the Session command may not be issued
        * again
        * <BR><PRE>
            Expected Major Result Code: SESSION_CREATED
        * </PRE><BR>
25        * @param pWholesalerAccountId the parameter value
        * @param pWholesalerPassword the parameter value
        * @param optionalAttributes a property list of optional attributes
        * @return the Wholesale Server Protocol Server response
        * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
30    unable to complete command
        * @see COMMAND_SESSION
        * @see PARM_WHOLESALE_ACCOUNT_ID
        * @see PARM_WHOLESALE_PASSWORD
        * @see WSPResponse#SESSION_CREATED
35        * @see WSPResponse#AUTHENTICATION_FAILED
        *
    */
    WSPResponse session(
        String pWholesalerAccountId,
40        String pWholesalerPassword,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return session( pWholesalerAccountId, pWholesalerPassword,
optionalAttributes, 0);
45    }
    protected WSPResponse session(
        String pWholesalerAccountId,
        String pWholesalerPassword,
50        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".session(String, String,
Properties, int) ";
        if (count > retry) {
55            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
    }

```

```

        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
5             conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

                // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);
10             allParmsAndAttrs.put(PARM_WHOLESALE_ACCOUNT_ID,
                pWholesalerAccountId);
                allParmsAndAttrs.put(PARM_WHOLESALE_PASSWORD,
                pWholesalerPassword);
15             returnCode = conn.makeRequest(COMMAND_SESSION,
                allParmsAndAttrs);
                pool.checkIn(conn);
                return returnCode;
20         } catch (Exception e) {
            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
25                 System.err.println(methodName + "Caught
                ResourceException" + re);
            }
            finally {
                return session(
30                     pWholesalerAccountId,
                        pWholesalerPassword,
                        optionalAttributes,
                        count + 1);
            }
        }
35     }

    /**
    * This method provides information about a specified target.
    * Currently, the only supported target is "Protocol".
    * This target tells Describe to return the current protocol version
    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
45     Expected Result Property Key(s) Returned:
        ATTR_PROTOCOL
    * </PRE><BR>
    * @param pTarget the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
50     unable to complete command
    * @see NSIWholesaleAPI#COMMAND_DESCRIBE
    * @see NSIWholesaleAPI#PARM_TARGET
    * @see NSIWholesaleAPI#ATTR_PROTOCOL
55     * @see
        WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *

```

```

    */
    public WSPResponse describe(String pTarget)
    throws NSIWholesaleException {
5        return describe(pTarget, new Properties(), 0);
    }
    /**
    * This method provides information about a specified target.
    * Currently, the only supported target is "Protocol".
    * This target tells Describe to return the current protocol version
10    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
        ATTR_PROTOCOL
15    * </PRE><BR>
    * @param pTarget the parameter value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
20    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_DESCRIBE
    * @see NSIWholesaleAPI#PARAM_TARGET
    * @see NSIWholesaleAPI#ATTR_PROTOCOL
    * @see
25    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    */
    public WSPResponse describe(String pTarget, Properties
    optionalAttributes)
30    throws NSIWholesaleException {
        return describe(pTarget, optionalAttributes, 0);
    }
    protected WSPResponse describe(
        String pTarget,
35        Properties optionalAttributes,
        int count)
    throws NSIWholesaleException {
        String methodName = className + ".describe(String, Properties,
40        int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
            complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;
45
        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();
50
            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(PARAM_TARGET, pTarget);
55
            returnCode = conn.makeRequest(COMMAND_DESCRIBE,
            allParmsAndAttrs);

```

```

        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
        try {
5           pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
10        }
        finally {
            return describe(
                pTarget,
                optionalAttributes,
15                count + 1);
        }
    }
}

20 /**
 * This method verifies whether or not a specified domain name is
available.
 * <BR><PRE>
    Expected Major Result Code:
25 COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 * </PRE><BR>
 * @param pDomainName the parameter value
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
30 unable to complete command
 * @see NSIWholesaleAPI#COMMAND_VERIFY
 * @see NSIWholesaleAPI#CLASS_DOMAIN_NAME
 * @see NSIWholesaleAPI#COMBO_VERIFY_DOMAIN_NAME
 * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
35 * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 *
 */
public WSPResponse verifyDomainName(
    String pDomainName)
40 throws NSIWholesaleException {
    return verifyDomainName(
        pDomainName,
        new Properties(),
        0);
45 }
/**
 * This method verifies whether or not a specified domain name is
available.
 * <BR><PRE>
50 Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 * </PRE><BR>
 * @param pDomainName the parameter value
 * @param optionalAttributes a property list of optional attributes
55 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command

```

```

    * @see NSIWholesaleAPI#COMMAND_VERIFY
    * @see NSIWholesaleAPI#CLASS_DOMAIN_NAME
    * @see NSIWholesaleAPI#COMBO_VERIFY_DOMAIN_NAME
    * @see NSIWholesaleAPI#PARAM_DOMAIN_NAME
5    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    *
    */
    public WSPResponse verifyDomainName(
        String pDomainName,
        Properties optionalAttributes)
10    throws NSIWholesaleException {
        return verifyDomainName(
            pDomainName,
            optionalAttributes,
15            0);
    }

    protected WSPResponse verifyDomainName(
        String pDomainName,
        Properties optionalAttributes,
        int count)
20    throws NSIWholesaleException {
        String methodName = className + ".verifyDomainName(String,
        Properties, int) ";
        if (count > retry) {
25            throw new NSIWholesaleException(methodName + "Unable to
        complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;
30
        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParamsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParamsAndAttrs, optionalAttributes);

            allParamsAndAttrs.put(PARAM_DOMAIN_NAME, pDomainName);
35
            returnCode = conn.makeRequest(COMBO_VERIFY_DOMAIN_NAME,
            allParamsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
45            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
50 ResourceException" + re);
            }
            finally {
                return verifyDomainName(
                    pDomainName,
                    optionalAttributes,
55                    count + 1);
            }
        }
    }

```

```

    }
}

/**
5   * This method verifies if series of domains are available. The command
   * will take the domain target string and add the .com, .net, and .org
   * TLD suffix and verify the domain name availability.
   * <BR><PRE>
       Expected Major Result Code:
10  COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
       Expected Result Property Key(s) Returned:
           DomainName1:
           DomainName2:
           DomainName3:
15  ...etc...
   * </PRE><BR>
   * @param pDomainTarget the parameter value
   * @return the Wholesale Server Protocol Server response
   * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
20  unable to complete command
   * @see NSIWholesaleAPI#COMMAND_VERIFY
   * @see NSIWholesaleAPI#CLASS_DOMAINS
   * @see NSIWholesaleAPI#COMBO_VERIFY_DOMAINS
   * @see NSIWholesaleAPI#PARAM_DOMAIN_TARGET
25  * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
   * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
   */
30  public WSPResponse verifyDomains(
       String pDomainTarget)
       throws NSIWholesaleException {
       return verifyDomains(
           pDomainTarget,
35  new Properties(),
           0);
   }
/**
40  * This method verifies if series of domains are available. The command
   * will take the domain target string and add the .com, .net, and .org
   * TLD suffix and verify the domain name availability.
   * <BR><PRE>
       Expected Major Result Code:
45  COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
       Expected Result Property Key(s) Returned:
           DomainName1:
           DomainName2:
           DomainName3:
50  ...etc...
   * </PRE><BR>
   * @param pDomainTarget the parameter value
   * @param optionalAttributes a property list of optional attributes
   * @return the Wholesale Server Protocol Server response
   * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
55  unable to complete command
   * @see NSIWholesaleAPI#COMMAND_VERIFY
   * @see NSIWholesaleAPI#CLASS_DOMAINS

```



```

        * @see NSIWholesaleAPI#COMBO_VERIFY_DOMAINS
        * @see NSIWholesaleAPI#PARAM_DOMAIN_TARGET
        * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
        * @see
5      WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        */
        public WSPResponse verifyDomains(
            String pDomainTarget,
10          Properties optionalAttributes)
            throws NSIWholesaleException {
            return verifyDomains(
                pDomainTarget,
                optionalAttributes,
15              0);
        }

        protected WSPResponse verifyDomains(
            String pDomainTarget,
            Properties optionalAttributes,
20          int count)
            throws NSIWholesaleException {
            String methodName = className + ".verifyDomains(String,
Properties, int) ";
            if (count > retry) {
25              throw new NSIWholesaleException(methodName + "Unable to
complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;
30          try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

35              // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);

                allParmsAndAttrs.put(PARAM_DOMAIN_TARGET, pDomainTarget);

40              returnCode = conn.makeRequest(COMBO_VERIFY_DOMAINS,
allParmsAndAttrs);
                pool.checkIn(conn);
                return returnCode;
            } catch (Exception e) {
45              try {
                    pool.release(conn);
                }
                catch (ResourceException re) {
                    System.err.println(methodName + "Caught
50 ResourceException" + re);
                }
                finally {
                    return verifyDomains(
80                  pDomainTarget,
                    optionalAttributes,
                    count + 1);
                }
            }
        }
    }

```

```

    }
}

/**
5   * This method verifies whether or not a specified customer account id
and  * password are valid.
    * <BR><PRE>
        Expected Major Result Code:
10  COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
    * @return the Wholesale Server Protocol Server response
15  * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_VERIFY
    * @see NSIWholesaleAPI#CLASS_PASSWORD
    * @see NSIWholesaleAPI#COMBO_VERIFY_PASSWORD
20  * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    *
    */
25  public WSPResponse verifyPassword(
        String pCustomerId,
        String pCustomerPassword)
    throws NSIWholesaleException {
        return verifyPassword(
30          pCustomerId,
          pCustomerPassword,
          new Properties(),
          0);
    }
35  /**
and  * This method verifies whether or not a specified customer account id
    * password are valid.
    * <BR><PRE>
40  Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
45  * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_VERIFY
50  * @see NSIWholesaleAPI#CLASS_PASSWORD
    * @see NSIWholesaleAPI#COMBO_VERIFY_PASSWORD
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
55  *
    */
    public WSPResponse verifyPassword(

```

```

        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes)
5      throws NSIWholesaleException {
        return verifyPassword(
            pCustomerId,
            pCustomerPassword,
            optionalAttributes,
            0);
10    }
    protected WSPResponse verifyPassword(
        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes,
15      int count)
        throws NSIWholesaleException {
        String methodName = className + ".verifyPassword(String, String,
Properties, int) ";
        - if -(count > retry) {
20            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;
25
        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
30      pCustomerId);
            allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
35      pCustomerPassword);

            returnCode = conn.makeRequest(COMBO_VERIFY_PASSWORD,
allParmsAndAttrs);
40            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
            try {
                pool.release(conn);
45            }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
ResourceException" + re);
50            }
            finally {
                return verifyPassword(
                    pCustomerId,
                    pCustomerPassword,
                    optionalAttributes,
55            count + 1);
            }
        }
    }

```

```

    }

    /**
5      * This method creates an individual customer account and returns an
      * automatically generated customer account id.
      * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
10      Expected Result Property Key(s) Returned:
        ATTR_CUSTOMER_ACCOUNT_ID
      * </PRE><BR>
      * @param aFirstName the attribute value
      * @param aLastName the attribute value
15      * @param aAddress1 the attribute value
      * @param aCountryCode the attribute value
      * @param aPhone the attribute value
      * @param aCustomerPassword the attribute value
      * @param aAuthQuestion the attribute value
20      * @param aAuthAnswer the attribute value
      * @param aCity the attribute value
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
25      * @see NSIWholesaleAPI#COMMAND_CREATE
      * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
      * @see NSIWholesaleAPI#COMBO_CREATE_INDIVIDUAL_ACCOUNT
      * @see NSIWholesaleAPI#ATTR_FIRST_NAME
      * @see NSIWholesaleAPI#ATTR_LAST_NAME
30      * @see NSIWholesaleAPI#ATTR_ADDRESS1
      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
      * @see NSIWholesaleAPI#ATTR_PHONE
      * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
35      * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
      * @see NSIWholesaleAPI#ATTR_CITY
      *
      * @see NSIWholesaleAPI#ATTR_STATE
      * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
40      * @see NSIWholesaleAPI#ATTR_ADDRESS2
      * @see NSIWholesaleAPI#ATTR_ADDRESS3
      * @see NSIWholesaleAPI#ATTR_ADDRESS4
      * @see NSIWholesaleAPI#ATTR_ADDRESS5
      * @see NSIWholesaleAPI#ATTR_EMAIL
45      * @see NSIWholesaleAPI#ATTR_FAX
      *
      * @see WSPResponse#INVALID_VALUE
      * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
50      * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
      *
    */
    public WSPResponse createIndividualAccount(
55        String aFirstName,
        String aLastName,
        String aAddress1,
        String aCountryCode,

```

```

        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
5         String aCity)
    throws NSIWholesaleException {
        return createIndividualAccount(
            aFirstName,
            aLastName,
10         aAddress1,
            aCountryCode,
            aPhone,
            aCustomerPassword,
            aAuthQuestion,
            aAuthAnswer,
15         aCity,
            new Properties(),
            0);
    }
    /**
    * This method creates an individual customer account and returns an
    * automatically generated customer account id.
    * <BR><PRE>
        Optional Attribute Key(s):
25         ATTR_ADDRESS2
        ATTR_ADDRESS3
        ATTR_ADDRESS4
        ATTR_ADDRESS5
        ATTR_EMAIL
30         ATTR_FAX

        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
35         ATTR_CUSTOMER_ACCOUNT_ID
    * </PRE><BR>
    * @param aFirstName the attribute value
    * @param aLastName the attribute value
    * @param aAddress1 the attribute value
40    * @param aCountryCode the attribute value
    * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
    * @param aAuthAnswer the attribute value
45    * @param aCity the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command
50    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_CREATE_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
55    * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE

```

```

    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_CITY
5
    *
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
    * @see NSIWholesaleAPI#ATTR_ADDRESS3
10
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_FAX
    *
15
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
    *
20
    */
    public WSPResponse createIndividualAccount(
        String aFirstName,
        String aLastName,
        String aAddress1,
25
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
30
        String aCity,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return createIndividualAccount(
            aFirstName,
35
            aLastName,
            aAddress1,
            aCountryCode,
            aPhone,
            aCustomerPassword,
40
            aAuthQuestion,
            aAuthAnswer,
            aCity,
            optionalAttributes,
            0);
45
    }
    protected WSPResponse createIndividualAccount(
        String aFirstName,
        String aLastName,
        String aAddress1,
50
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
55
        String aCity,
        Properties optionalAttributes,
        int count)

```

```

        throws NSIWholesaleException {
            String methodName = className +
".createIndividualAccount(StringX9, Properties, int) ";
            if (count > retry) {
5              throw new NSIWholesaleException(methodName + "Unable to
complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;
10
            try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

15                // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);

                allParmsAndAttrs.put(ATTR_FIRST_NAME, aFirstName);
                allParmsAndAttrs.put(ATTR_LAST_NAME, aLastName);
20                allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
                allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
                allParmsAndAttrs.put(ATTR_PHONE, aPhone);
                allParmsAndAttrs.put(ATTR_CUSTOMER_PASSWORD,
aCustomerPassword);
25                allParmsAndAttrs.put(ATTR_AUTH_QUESTION, aAuthQuestion);
                allParmsAndAttrs.put(ATTR_AUTH_ANSWER, aAuthAnswer);
                allParmsAndAttrs.put(ATTR_CITY, aCity);

                returnCode =
30                conn.makeRequest(COMBO_CREATE_INDIVIDUAL_ACCOUNT, allParmsAndAttrs);
                pool.checkIn(conn);
                return returnCode;
            }
            catch (Exception e) {
35                try {
                    pool.release(conn);
                }
                catch (ResourceException re) {
                    System.err.println(methodName + "Caught
40 ResourceException" + re);
                }
                finally {
                    return createIndividualAccount(
45                        aFirstName,
                        aLastName,
                        aAddress1,
                        aCountryCode,
                        aPhone,
                        aCustomerPassword,
50                        aAuthQuestion,
                        aAuthAnswer,
                        aCity,
                        optionalAttributes,
                        count + 1);
55                }
            }
        }
    }

```

```

/**
 * This method creates an individual customer account and returns an
 * automatically generated customer account id.
5  * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
    ATTR_CUSTOMER_ACCOUNT_ID
10  * </PRE><BR>
    * @param aFirstName the attribute value
    * @param aLastName the attribute value
    * @param aAddress1 the attribute value
    * @param aCountryCode the attribute value
15  * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
    * @param aAuthAnswer the attribute value
    * @param aCity the attribute value
20  * @param aState the attribute value
    * @param aPostalCode the attribute value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
25  * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_CREATE_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
30  * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
35  * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    *
40  * @see NSIWholesaleAPI#ATTR_ADDRESS2
    * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
45  * @see NSIWholesaleAPI#ATTR_FAX
    *
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
50  * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
    *
 */
public WSPResponse createIndividualAccount(
55  String aFirstName,
    String aLastName,
    String aAddress1,
    String aCountryCode,

```



```

        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
5         String aCity,
        String aState,
        String aPostalCode)
throws NSIWholesaleException {
        return createIndividualAccount(
10         aFirstName,
        aLastName,
        aAddress1,
        aCountryCode,
        aPhone,
15         aCustomerPassword,
        aAuthQuestion,
        aAuthAnswer,
        aCity,
        aState,
20         aPostalCode,
        new Properties(),
        0);
    }
    /**
25     * This method creates an individual customer account and returns an
    * automatically generated customer account id.
    * <BR><PRE>
        Optional Attribute Key(s):
            ATTR_ADDRESS2
30            ATTR_ADDRESS3
            ATTR_ADDRESS4
            ATTR_ADDRESS5
            ATTR_EMAIL
            ATTR_FAX
35
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_CUSTOMER_ACCOUNT_ID
40     * </PRE><BR>
    * @param aFirstName the attribute value
    * @param aLastName the attribute value
    * @param aAddress1 the attribute value
    * @param aCountryCode the attribute value
45     * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
    * @param aAuthAnswer the attribute value
    * @param aCity the attribute value
50     * @param aState the attribute value
    * @param aPostalCode the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
55 unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT

```

```

    * @see NSIWholesaleAPI#COMBO_CREATE_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_ADDRESS1
5      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
10     * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    *
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
15     * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_FAX
20     *
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
25     * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
    *
    */
    public WSPResponse createIndividualAccount(
        String aFirstName,
        String aLastName,
30     String aAddress1,
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
35     String aAuthAnswer,
        String aCity,
        String aState,
        String aPostalCode,
        Properties optionalAttributes)
40     throws NSIWholesaleException {
        return createIndividualAccount(
            aFirstName,
            aLastName,
            aAddress1,
45     aCountryCode,
            aPhone,
            aCustomerPassword,
            aAuthQuestion,
            aAuthAnswer,
50     aCity,
            aState,
            aPostalCode,
            optionalAttributes,
            0);
55     }
    protected WSPResponse createIndividualAccount(
        String aFirstName,

```

```

        String aLastName,
        String aAddress1,
        String aCountryCode,
        String aPhone,
5         String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
        String aCity,
        String aState,
10        String aPostalCode,
        Properties optionalAttributes,
        int count)
    throws NSIWholesaleException {
        String methodName = className +
15        ".createIndividualAccount(StringX11, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
20        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
25            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(ATTR_FIRST_NAME, aFirstName);
            allParmsAndAttrs.put(ATTR_LAST_NAME, aLastName);
            allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
            allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
            allParmsAndAttrs.put(ATTR_PHONE, aPhone);
35            allParmsAndAttrs.put(ATTR_CUSTOMER_PASSWORD,
aCustomerPassword);
            allParmsAndAttrs.put(ATTR_AUTH_QUESTION, aAuthQuestion);
            allParmsAndAttrs.put(ATTR_AUTH_ANSWER, aAuthAnswer);
            allParmsAndAttrs.put(ATTR_CITY, aCity);
40            allParmsAndAttrs.put(ATTR_STATE, aState);
            allParmsAndAttrs.put(ATTR_POSTAL_CODE, aPostalCode);

            returnCode =
conn.makeRequest(COMBO_CREATE_INDIVIDUAL_ACCOUNT, allParmsAndAttrs);
45            pool.checkIn(conn);
            return returnCode;
        }
        catch (Exception e) {
            try {
50                pool.release(conn);
            }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
ResourceException" + re);
55            }
            finally {
                return createIndividualAccount(

```

```

5         aFirstName,
        aLastName,
        aAddress1,
        aCountryCode,
        aPhone,
        aCustomerPassword,
        aAuthQuestion,
        aAuthAnswer,
10        aCity,
        aState,
        aPostalCode,
        optionalAttributes,
        count + 1);
    }
15    }
}

/**
 * This method creates a business customer account and returns an
 * automatically generated customer account id.
 * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
25    ATTR_CUSTOMER_ACCOUNT_ID
    * </PRE><BR>
    * @param aCompanyName the attribute value
    * @param aCompanyType the attribute value
    * @param aAddress1 the attribute value
30    * @param aCity the attribute value
    * @param aCountryCode the attribute value
    * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
35    * @param aAuthAnswer the attribute value
    * @param aLegalContactFirstName the attribute value
    * @param aLegalContactLastName the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
40    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_CREATE_BUSINESS_ACCOUNT
45    * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
    * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
    * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
50    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
55    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
    *
    * @see NSIWholesaleAPI#ATTR_STATE

```

```

    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
    * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
5    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
10   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
15   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_EMAIL
20   *
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
25   *
    */
    public WSPResponse createBusinessAccount(
        String aCompanyName,
        String aCompanyType,
30     String aAddress1,
        String aCity,
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
35     String aAuthAnswer,
        String aLegalContactFirstName,
        String aLegalContactLastName)
    throws NSIWholesaleException {
40     return createBusinessAccount(
        aCompanyName,
        aCompanyType,
        aAddress1,
        aCity,
45     aCountryCode,
        aPhone,
        aCustomerPassword,
        aAuthQuestion,
        aAuthAnswer,
50     aLegalContactFirstName,
        aLegalContactLastName,
        new Properties(),
        0);
    }
55   /**
    * This method creates a business customer account and returns an
    * automatically generated customer account id.

```

```

* <BR><PRE>
  Optional Attribute Key(s):
    ATTR_ADDRESS2
    ATTR_ADDRESS3
5    ATTR_ADDRESS4
    ATTR_ADDRESS5
    ATTR_EMAIL
    ATTR_FAX
10   ATTR_LEGAL_CONTACT_ADDRESS1
    ATTR_LEGAL_CONTACT_ADDRESS2
    ATTR_LEGAL_CONTACT_ADDRESS3
    ATTR_LEGAL_CONTACT_ADDRESS4
    ATTR_LEGAL_CONTACT_ADDRESS5
15   ATTR_LEGAL_CONTACT_CITY
    ATTR_LEGAL_CONTACT_STATE
    ATTR_LEGAL_CONTACT_POSTAL_CODE
    ATTR_LEGAL_CONTACT_COUNTRY_CODE
    ATTR_LEGAL_CONTACT_PHONE
20   ATTR_LEGAL_CONTACT_FAX
    ATTR_LEGAL_CONTACT_EMAIL

    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
25   ATTR_CUSTOMER_ACCOUNT_ID

* </PRE><BR>
* @param aCompanyName the attribute value
* @param aCompanyType the attribute value
* @param aAddress1 the attribute value
30  * @param aCity the attribute value
    * @param aCountryCode the attribute value
    * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
35  * @param aAuthAnswer the attribute value
    * @param aLegalContactFirstName the attribute value
    * @param aLegalContactLastName the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
40  * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_CREATE_BUSINESS_ACCOUNT
45  * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
    * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
    * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_CITY
50  * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
55  * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
    *
    * @see NSIWholesaleAPI#ATTR_STATE

```

```

    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
    * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
5    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
10   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
15   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_EMAIL
20   *
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
25   *
    */
    public WSPResponse createBusinessAccount(
        String aCompanyName,
        String aCompanyType,
30     String aAddress1,
        String aCity,
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
        String aLegalContactFirstName,
        String aLegalContactLastName,
        Properties optionalAttributes)
40     throws NSIWholesaleException {
        return createBusinessAccount(
            aCompanyName,
            aCompanyType,
            aAddress1,
45     aCity,
            aCountryCode,
            aPhone,
            aCustomerPassword,
            aAuthQuestion,
            aAuthAnswer,
50     aLegalContactFirstName,
            aLegalContactLastName,
            optionalAttributes,
            0);
55     }
    protected WSPResponse createBusinessAccount(
        String aCompanyName,

```

```

        String aCompanyType,
        String aAddress1,
        String aCity,
        String aCountryCode,
5         String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
10        String aLegalContactFirstName,
        String aLegalContactLastName,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".createBusinessAccount(StringX11,
15        Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
25            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

30            allParmsAndAttrs.put(ATTR_COMPANY_NAME, aCompanyName);
            allParmsAndAttrs.put(ATTR_COMPANY_TYPE, aCompanyType);
            allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
            allParmsAndAttrs.put(ATTR_CITY, aCity);
            allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
35            allParmsAndAttrs.put(ATTR_PHONE, aPhone);
            allParmsAndAttrs.put(ATTR_CUSTOMER_PASSWORD,
aCustomerPassword);
            allParmsAndAttrs.put(ATTR_AUTH_QUESTION, aAuthQuestion);
            allParmsAndAttrs.put(ATTR_AUTH_ANSWER, aAuthAnswer);
40            allParmsAndAttrs.put(ATTR_LEGAL_CONTACT_FIRST_NAME,
aLegalContactFirstName);
            allParmsAndAttrs.put(ATTR_LEGAL_CONTACT_LAST_NAME,
aLegalContactLastName);

45            returnCode = conn.makeRequest(COMBO_CREATE_BUSINESS_ACCOUNT,
allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
50            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
55 ResourceException" + re);
            }
            finally {

```



```

        return createBusinessAccount (
            aCompanyName,
            aCompanyType,
            aAddress1,
5           aCity,
            aCountryCode,
            aPhone,
            aCustomerPassword,
            aAuthQuestion,
10          aAuthAnswer,
            aLegalContactFirstName,
            aLegalContactLastName,
            optionalAttributes,
            count + 1);
15          }
        }
    }

/**
20  * This method creates a business customer account and returns an
    * automatically generated customer account id.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
25     Expected Result Property Key(s) Returned:
        ATTR_CUSTOMER_ACCOUNT_ID
    * </PRE><BR>
    * @param aCompanyName the attribute value
    * @param aCompanyType the attribute value
30  * @param aAddress1 the attribute value
    * @param aCity the attribute value
    * @param aState the attribute value
    * @param aPostalCode the attribute value
    * @param aCountryCode the attribute value
35  * @param aPhone the attribute value
    * @param aCustomerPassword the attribute value
    * @param aAuthQuestion the attribute value
    * @param aAuthAnswer the attribute value
    * @param aLegalContactFirstName the attribute value
40  * @param aLegalContactLastName the attribute value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
45  * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_CREATE_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
    * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
50  * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
55  * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER

```

```

    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
    *
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
5    * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_FAX
10   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
15   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
20   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_EMAIL
    *
    * @see WSPResponse#INVALID_VALUE
    * @see
25   WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
    *
    */
    public WSPResponse createBusinessAccount(
30        String aCompanyName,
        String aCompanyType,
        String aAddress1,
        String aCity,
        String aCountryCode,
35        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
        String aLegalContactFirstName,
40        String aLegalContactLastName,
        String aState,
        String aPostalCode)
        throws NSIWholesaleException {
        return createBusinessAccount(
45            aCompanyName,
            aCompanyType,
            aAddress1,
            aCity,
            aCountryCode,
50            aPhone,
            aCustomerPassword,
            aAuthQuestion,
            aAuthAnswer,
            aLegalContactFirstName,
55            aLegalContactLastName,
            aState,
            aPostalCode,

```

```

        new Properties(),
        0);
    }
    /**
5      * This method creates a business customer account and returns an
      * automatically generated customer account id.
      * <BR><PRE>
        Optional Attribute Key(s):
10          ATTR_ADDRESS2
          ATTR_ADDRESS3
          ATTR_ADDRESS4
          ATTR_ADDRESS5
          ATTR_EMAIL
          ATTR_FAX
15          ATTR_LEGAL_CONTACT_ADDRESS1
          ATTR_LEGAL_CONTACT_ADDRESS2
          ATTR_LEGAL_CONTACT_ADDRESS3
          ATTR_LEGAL_CONTACT_ADDRESS4
          ATTR_LEGAL_CONTACT_ADDRESS5
20          ATTR_LEGAL_CONTACT_CITY
          ATTR_LEGAL_CONTACT_STATE
          ATTR_LEGAL_CONTACT_POSTAL_CODE
          ATTR_LEGAL_CONTACT_COUNTRY_CODE
          ATTR_LEGAL_CONTACT_PHONE
25          ATTR_LEGAL_CONTACT_FAX
          ATTR_LEGAL_CONTACT_EMAIL

        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
30      Expected Result Property Key(s) Returned:
          ATTR_CUSTOMER_ACCOUNT_ID
      * </PRE><BR>
      * @param aCompanyName the attribute value
      * @param aCompanyType the attribute value
35      * @param aAddress1 the attribute value
      * @param aCity the attribute value
      * @param aState the attribute value
      * @param aPostalCode the attribute value
      * @param aCountryCode the attribute value
40      * @param aPhone the attribute value
      * @param aCustomerPassword the attribute value
      * @param aAuthQuestion the attribute value
      * @param aAuthAnswer the attribute value
      * @param aLegalContactFirstName the attribute value
45      * @param aLegalContactLastName the attribute value
      * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
50      unable to complete command
      * @see NSIWholesaleAPI#COMMAND_CREATE
      * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
      * @see NSIWholesaleAPI#COMBO_CREATE_BUSINESS_ACCOUNT
      * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
      * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
55      * @see NSIWholesaleAPI#ATTR_ADDRESS1
      * @see NSIWholesaleAPI#ATTR_CITY
      * @see NSIWholesaleAPI#ATTR_STATE

```

```

5      * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
      * @see NSIWholesaleAPI#ATTR_PHONE
      * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
      * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
      *
10     * @see NSIWholesaleAPI#ATTR_ADDRESS2
      * @see NSIWholesaleAPI#ATTR_ADDRESS3
      * @see NSIWholesaleAPI#ATTR_ADDRESS4
      * @see NSIWholesaleAPI#ATTR_ADDRESS5
      * @see NSIWholesaleAPI#ATTR_EMAIL
15     * @see NSIWholesaleAPI#ATTR_FAX
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
20     * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
25     * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
      * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_EMAIL
      *
      * @see WSPResponse#INVALID_VALUE
30     * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
      * @see NSIWholesaleAPI#ATTR_CUSTOMER_ACCOUNT_ID
      *
*/
35     public WSPResponse createBusinessAccount(
        String aCompanyName,
        String aCompanyType,
        String aAddress1,
        String aCity,
40         String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
        String aAuthAnswer,
45         String aLegalContactFirstName,
        String aLegalContactLastName,
        String aState,
        String aPostalCode,
        Properties optionalAttributes)
50     throws NSIWholesaleException {
        return createBusinessAccount(
            aCompanyName,
            aCompanyType,
            aAddress1,
55         aCity,
            aCountryCode,
            aPhone,

```

```

        aCustomerPassword,
        aAuthQuestion,
        aAuthAnswer,
        aLegalContactFirstName,
5         aLegalContactLastName,
        aState,
        aPostalCode,
        optionalAttributes,
        0);
10     }
    protected WSPResponse createBusinessAccount(
        String aCompanyName,
        String aCompanyType,
        String aAddress1,
15         String aCity,
        String aCountryCode,
        String aPhone,
        String aCustomerPassword,
        String aAuthQuestion,
20         String aAuthAnswer,
        String aLegalContactFirstName,
        String aLegalContactLastName,
        String aState,
        String aPostalCode,
25         Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".createBusinessAccount(StringX13,
Properties, int) ";
30         if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
35         WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();
40
            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(ATTR_COMPANY_NAME, aCompanyName);
            allParmsAndAttrs.put(ATTR_COMPANY_TYPE, aCompanyType);
45         allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
            allParmsAndAttrs.put(ATTR_CITY, aCity);
            allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
            allParmsAndAttrs.put(ATTR_PHONE, aPhone);
            allParmsAndAttrs.put(ATTR_CUSTOMER_PASSWORD,
50         aCustomerPassword);
            allParmsAndAttrs.put(ATTR_AUTH_QUESTION, aAuthQuestion);
            allParmsAndAttrs.put(ATTR_AUTH_ANSWER, aAuthAnswer);
            allParmsAndAttrs.put(ATTR_LEGAL_CONTACT_FIRST_NAME,
55         aLegalContactFirstName);
            allParmsAndAttrs.put(ATTR_LEGAL_CONTACT_LAST_NAME,
            aLegalContactLastName);

```

```

        allParmsAndAttrs.put(ATTR_STATE, aState);
        allParmsAndAttrs.put(ATTR_POSTAL_CODE, aPostalCode);

        returnCode = conn.makeRequest(COMBO_CREATE_BUSINESS_ACCOUNT,
5    allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
        try {
10            pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
15        }
        finally {
            return createBusinessAccount(
                aCompanyName,
                aCompanyType,
20            aAddress1,
                aCity,
                aCountryCode,
                aPhone,
                aCustomerPassword,
25            aAuthQuestion,
                aAuthAnswer,
                aLegalContactFirstName,
                aLegalContactLastName,
                aState,
                aPostalCode,
                optionalAttributes,
                count + 1);
            }
        }
35    }

/**
 * This method creates a technical contact and returns an
 * automatically generated technical contact id (NIC Handle).
 * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
45    ATTR_TECH_CONTACT_ID
 * </PRE><BR>
 * @param aFirstName the attribute value
 * @param aLastName the attribute value
 * @param aOrganization the attribute value
50 * @param aAddress1 the attribute value
 * @param aCountryCode the attribute value
 * @param aPhone the attribute value
 * @param aEmail the attribute value
 * @param aTechContactPassword the attribute value
55 * @param aCity the attribute value
 * @return the Wholesale Server Protocol Server response

```

```

    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
5    * @see NSIWholesaleAPI#COMBO_CREATE_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_ORGANIZATION
    * @see NSIWholesaleAPI#ATTR_ADDRESS1
10    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_PASSWORD
    * @see NSIWholesaleAPI#ATTR_CITY
15    *
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_FAX
    *
20    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
25    */
    public WSPResponse createTechnicalContact(
        String aFirstName,
        String aLastName,
        String aOrganization,
30        String aAddress1,
        String aCountryCode,
        String aPhone,
        String aEmail,
        String aTechContactPassword,
        String aCity)
35    throws NSIWholesaleException {
        return createTechnicalContact(
            aFirstName,
            aLastName,
40            aOrganization,
            aAddress1,
            aCountryCode,
            aPhone,
            aEmail,
            aTechContactPassword,
            aCity,
45            new Properties(),
            0);
    }
50    /**
    * This method creates a technical contact and returns an
    * automatically generated technical contact id (NIC Handle).
    * <BR><PRE>
        Optional Attribute Key(s):
55        ATTR_FAX

```

```

Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
Expected Result Property Key(s) Returned:
ATTR_TECH_CONTACT_ID
5      * </PRE><BR>
      * @param aFirstName the attribute value
      * @param aLastName the attribute value
      * @param aOrganization the attribute value
      * @param aAddress1 the attribute value
10     * @param aCountryCode the attribute value
      * @param aPhone the attribute value
      * @param aEmail the attribute value
      * @param aTechContactPassword the attribute value
      * @param aCity the attribute value
15     * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
      * @see NSIWholesaleAPI#COMMAND_CREATE
20     * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
      * @see NSIWholesaleAPI#COMBO_CREATE_TECHNICAL_CONTACT
      * @see NSIWholesaleAPI#ATTR_FIRST_NAME
      * @see NSIWholesaleAPI#ATTR_LAST_NAME
      * @see NSIWholesaleAPI#ATTR_ORGANIZATION
25     * @see NSIWholesaleAPI#ATTR_ADDRESS1
      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
      * @see NSIWholesaleAPI#ATTR_PHONE
      * @see NSIWholesaleAPI#ATTR_EMAIL
      * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_PASSWORD
30     * @see NSIWholesaleAPI#ATTR_CITY
      *
      * @see NSIWholesaleAPI#ATTR_STATE
      * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
      * @see NSIWholesaleAPI#ATTR_FAX
35     *
      * @see WSPResponse#INVALID_VALUE
      * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
40     * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
      *
      */
      public WSPResponse createTechnicalContact(
          String aFirstName,
          String aLastName,
45         String aOrganization,
          String aAddress1,
          String aCountryCode,
          String aPhone,
          String aEmail,
50         String aTechContactPassword,
          String aCity,
          Properties optionalAttributes)
      throws NSIWholesaleException {
          return createTechnicalContact(
55             aFirstName,
             aLastName,
             aOrganization,

```



```

        aAddress1,
        aCountryCode,
        aPhone,
        aEmail,
5        aTechContactPassword,
        aCity,
        optionalAttributes,
        0);
    }
10    protected WSPResponse createTechnicalContact(
        String aFirstName,
        String aLastName,
        String aOrganization,
        String aAddress1,
15        String aCountryCode,
        String aPhone,
        String aEmail,
        String aTechContactPassword,
        String aCity,
20        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".createTechnicalContact(StringX9,
Properties, int) ";
25        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
30        WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();
35
            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(ATTR_FIRST_NAME, aFirstName);
            allParmsAndAttrs.put(ATTR_LAST_NAME, aLastName);
40            allParmsAndAttrs.put(ATTR_ORGANIZATION, aOrganization);
            allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
            allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
            allParmsAndAttrs.put(ATTR_PHONE, aPhone);
            allParmsAndAttrs.put(ATTR_EMAIL, aEmail);
45            allParmsAndAttrs.put(ATTR_TECH_CONTACT_PASSWORD,
aTechContactPassword);
            allParmsAndAttrs.put(ATTR_CITY, aCity);

            returnCode =
50            conn.makeRequest(COMBO_CREATE_TECHNICAL_CONTACT, allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
55            try {
                pool.release(conn);
            }

```

```

        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
        }
5         finally {
            return createTechnicalContact(
                aFirstName,
                aLastName,
                aOrganization,
10                aAddress1,
                aCountryCode,
                aPhone,
                aEmail,
                aTechContactPassword,
15                aCity,
                optionalAttributes,
                count + 1);
        }
    }
20 }

/**
 * This method creates a technical contact and returns an
 * automatically generated technical contact id (NIC Handle).
25 * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
    ATTR_TECH_CONTACT_ID
30 * </PRE><BR>
 * @param aFirstName the attribute value
 * @param aLastName the attribute value
 * @param aOrganization the attribute value
 * @param aAddress1 the attribute value
35 * @param aCountryCode the attribute value
 * @param aPhone the attribute value
 * @param aEmail the attribute value
 * @param aTechContactPassword the attribute value
 * @param aCity the attribute value
40 * @param aState the attribute value
 * @param aPostalCode the attribute value
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
45 * @see NSIWholesaleAPI#COMMAND_CREATE
 * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
 * @see NSIWholesaleAPI#COMBO_CREATE_TECHNICAL_CONTACT
 * @see NSIWholesaleAPI#ATTR_FIRST_NAME
 * @see NSIWholesaleAPI#ATTR_LAST_NAME
50 * @see NSIWholesaleAPI#ATTR_ORGANIZATION
 * @see NSIWholesaleAPI#ATTR_ADDRESS1
 * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
 * @see NSIWholesaleAPI#ATTR_PHONE
 * @see NSIWholesaleAPI#ATTR_EMAIL
55 * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_PASSWORD
 * @see NSIWholesaleAPI#ATTR_CITY
 * @see NSIWholesaleAPI#ATTR_STATE

```

```

    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    *
    * @see NSIWholesaleAPI#ATTR_FAX
    *
5    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
10   */
   public WSPResponse createTechnicalContact(
       String aFirstName,
       String aLastName,
       String aOrganization,
15      String aAddress1,
       String aCountryCode,
       String aPhone,
       String aEmail,
       String aTechContactPassword,
20      String aCity,
       String aState,
       String aPostalCode)
   throws NSIWholesaleException {
       return createTechnicalContact(
25          aFirstName,
          aLastName,
          aOrganization,
          aAddress1,
          aCountryCode,
30          aPhone,
          aEmail,
          aTechContactPassword,
          aCity,
          aState,
35          aPostalCode,
          new Properties(),
          0);
   }
   /**
40   * This method creates a technical contact and returns an
   * automatically generated technical contact id (NIC Handle).
   * <BR><PRE>
       Optional Attribute Key(s):
           ATTR_FAX
45
       Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
       Expected Result Property Key(s) Returned:
           ATTR_TECH_CONTACT_ID
50   * </PRE><BR>
   * @param aFirstName the attribute value
   * @param aLastName the attribute value
   * @param aOrganization the attribute value
   * @param aAddress1 the attribute value
55   * @param aCountryCode the attribute value
   * @param aPhone the attribute value
   * @param aEmail the attribute value

```

```

    * @param aTechContactPassword the attribute value
    * @param aCity the attribute value
    * @param aState the attribute value
    * @param aPostalCode the attribute value
5    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
10    * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#COMBO_CREATE_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_ORGANIZATION
15    * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_PASSWORD
-20    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    *
    * @see NSIWholesaleAPI#ATTR_FAX
25    *
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
30    *
    */
    public WSPResponse createTechnicalContact(
        String aFirstName,
        String aLastName,
35        String aOrganization,
        String aAddress1,
        String aCountryCode,
        String aPhone,
        String aEmail,
40        String aTechContactPassword,
        String aCity,
        String aState,
        String aPostalCode,
        Properties optionalAttributes)
45    throws NSIWholesaleException {
        return createTechnicalContact(
            aFirstName,
            aLastName,
            aOrganization,
50            aAddress1,
            aCountryCode,
            aPhone,
            aEmail,
            aTechContactPassword,
            aCity,
55            aState,
            aPostalCode,

```

```

        optionalAttributes,
        0);
    }
    protected WSPResponse createTechnicalContact(
5        String aFirstName,
        String aLastName,
        String aOrganization,
        String aAddress1,
        String aCountryCode,
10        String aPhone,
        String aEmail,
        String aTechContactPassword,
        String aCity,
        String aState,
15        String aPostalCode,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className +
20        ".createTechnicalContact(StringX11, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
25        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

35            allParmsAndAttrs.put(ATTR_FIRST_NAME, aFirstName);
            allParmsAndAttrs.put(ATTR_LAST_NAME, aLastName);
            allParmsAndAttrs.put(ATTR_ORGANIZATION, aOrganization);
            allParmsAndAttrs.put(ATTR_ADDRESS1, aAddress1);
            allParmsAndAttrs.put(ATTR_COUNTRY_CODE, aCountryCode);
40            allParmsAndAttrs.put(ATTR_PHONE, aPhone);
            allParmsAndAttrs.put(ATTR_EMAIL, aEmail);
            allParmsAndAttrs.put(ATTR_TECH_CONTACT_PASSWORD,
aTechContactPassword);
            allParmsAndAttrs.put(ATTR_CITY, aCity);
            allParmsAndAttrs.put(ATTR_STATE, aState);
45            allParmsAndAttrs.put(ATTR_POSTAL_CODE, aPostalCode);

            returnCode =
            conn.makeRequest(COMBO_CREATE_TECHNICAL_CONTACT, allParmsAndAttrs);
50            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
            try {
                pool.release(conn);
55            }
            catch (ResourceException re) {

```

```

        System.err.println(methodName + "Caught
ResourceException" + re);
    }
    finally {
5         return createTechnicalContact(
                aFirstName,
                aLastName,
                aOrganization,
                aAddress1,
10             aCountryCode,
                aPhone,
                aEmail,
                aTechContactPassword,
                aCity,
15             aState,
                aPostalCode,
                optionalAttributes,
                count + 1);
        }
    }
}

/**
25  * This method registers a new domain product for an existing
  * customer account. This request assumes that the domain
  * name will be hosted by some organization other than
  * Network Solutions.
  * <BR>NOTE: This command is asynchronous. Receipt of the
30  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
  * indicates that the command has been successfully queued.
  * <BR><PRE>
      Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
35      Expected Result Property Key(s) Returned:
          ATTR_ISP_PASSWORD
  * </PRE><BR>
  * @param pCustomerId the parameter value
  * @param pCustomerPassword the parameter value
40  * @param aDomainName the attribute value
  * @param aHostName1 the attribute value
  * @param aHostAddr1 the attribute value
  * @param aHostName2 the attribute value
  * @param aHostAddr2 the attribute value
45  * @param aTechContactId the attribute value
  * @return the Wholesale Server Protocol Server response
  * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
  * @see NSIWholesaleAPI#COMMAND_CREATE
50  * @see NSIWholesaleAPI#CLASS_DOMAIN
  * @see NSIWholesaleAPI#COMBO_CREATE_DOMAIN
  * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
  * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
  * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
55  * @see NSIWholesaleAPI#ATTR_HOST_NAME1
  * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
  * @see NSIWholesaleAPI#ATTR_HOST_NAME2

```

```

    * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
5    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
10    */
    public WSPResponse createDomain(
        String pCustomerId,
        String pCustomerPassword,
15        String aDomainName,
        String aHostName1,
        String aHostAddr1,
        String aHostName2,
        String aHostAddr2,
20        String aTechContactId
    )
    throws NSIWholesaleException {
        return createDomain(
            pCustomerId,
25            pCustomerPassword,
            aDomainName,
            aHostName1,
            aHostAddr1,
            aHostName2,
30            aHostAddr2,
            aTechContactId,
            new Properties(),
            0);
    }
35    /**
    * This method registers a new domain product for an existing
    * customer account. This request assumes that the domain
    * name will be hosted by some organization other than
    * Network Solutions.
40    * <BR>NOTE: This command is asynchronous. Receipt of the
    * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
45    COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_ISP_PASSWORD
    * </PRE><BR>
    * @param pCustomerId the parameter value
50    * @param pCustomerPassword the parameter value
    * @param aDomainName the attribute value
    * @param aHostName1 the attribute value
    * @param aHostAddr1 the attribute value
    * @param aHostName2 the attribute value
55    * @param aHostAddr2 the attribute value
    * @param aTechContactId the attribute value
    * @param optionalAttributes a property list of optional attributes

```

```

    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
5      * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_CREATE_DOMAIN
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
10     * @see NSIWholesaleAPI#ATTR_HOST_NAME1
    * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
    * @see NSIWholesaleAPI#ATTR_HOST_NAME2
    * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
15     *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see
20     WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
    *
    */
    public WSPResponse createDomain(
25        String pCustomerId,
        String pCustomerPassword,
        String aDomainName,
        String aHostName1,
        String aHostAddr1,
30        String aHostName2,
        String aHostAddr2,
        String aTechContactId,
        Properties optionalAttributes)
    throws NSIWholesaleException {
35        return createDomain(
            pCustomerId,
            pCustomerPassword,
            aDomainName,
            aHostName1,
40            aHostAddr1,
            aHostName2,
            aHostAddr2,
            aTechContactId,
            optionalAttributes,
45            0);
    }

    protected WSPResponse createDomain(
        String pCustomerId,
        String pCustomerPassword,
50        String aDomainName,
        String aHostName1,
        String aHostAddr1,
        String aHostName2,
        String aHostAddr2,
        String aTechContactId,
60        Properties optionalAttributes,
        int count)

```



```

        throws NSIWholesaleException {
            int rc;
            String methodName = className + ".createDomain(StringX8,
5      Properties, int) ";
            System.out.println("*** Made it into createDomain...");
            if (count > retry) {
                rc = refundCard(pCustomerAccountId, aDomainName,
                                CreateDomainPrice, CreateDomainWNRP,
                                optionalAttributes);
10      throw new NSIWholesaleException(methodName + "Unable to
complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;
15      // Double check to see if the domain really is available (to
minimize charges)
            // with immediate credits
            try {
                System.out.println("***Attempting to verify domain is
20      available.");
                WSPResponse wspResponse = verifyDomainName( aDomainName);
                if (wspResponse.getMajorResultCode() ==
WSPResponse.COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED) {
25      // See if credit card information is available, if so get
transaction
                System.out.println("*** Begining to do Charge Card
Validation.");
                rc = ChargeCard(pCustomerAccountId, aDomainName,
                                CreateDomainPrice, CreateDomainWNRP,
                                optionalAttributes);
30      if (rc != 0) {
                // Bad credit card. Return bad code.
                throw new NSIWholesaleException(methodName + "Credit card
35      declined for:" +
                                aDomainName);
            }
            } else {
                // Domain name not available. Return bad code.
40      throw new NSIWholesaleException(methodName + "Domain name
not available:"+
                                aDomainName);
            }
        }
        catch (NSIWholesaleException e) {
45      throw new NSIWholesaleException("Domain name previously
taken:" + aDomainName);
        }
        // All o.k. create the domain.
50      System.out.println("*** After the credit card interaction,
going on to create a name.");
        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();
55      // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

```

```

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
5 pCustomerPassword);
        allParmsAndAttrs.put(ATTR_DOMAIN_NAME, aDomainName);
        allParmsAndAttrs.put(ATTR_HOST_NAME1, aHostName1);
        allParmsAndAttrs.put(ATTR_HOST_ADDR1, aHostAddr1);
        allParmsAndAttrs.put(ATTR_HOST_NAME2, aHostName2);
10 allParmsAndAttrs.put(ATTR_HOST_ADDR2, aHostAddr2);
        allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);

        returnCode = conn.makeRequest(COMBO_CREATE_DOMAIN,
allParmsAndAttrs);
15 pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
        try {
            pool.release(conn);
20        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
        }
25        finally {
            return createDomain(
                pCustomerAccountId,
                pCustomerPassword,
                aDomainName,
30                aHostName1,
                aHostAddr1,
                aHostName2,
                aHostAddr2,
                aTechContactId,
                optionalAttributes,
                count + 1);
35        }
    }
}

40
/**
 * This method registers a new domain product for an existing
 * customer account. This request assumes that the domain
 * name will be hosted by some organization other than
 * Network Solutions.
45 * <BR>NOTE: This command is asynchronous. Receipt of the
 * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.
 * <BR><PRE>
50     Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
        ATTR_ISP_PASSWORD
    * </PRE><BR>
55 * @param pCustomerAccountId the parameter value
 * @param pCustomerPassword the parameter value
 * @param aDomainName the attribute value

```

```

    * @param aHostId1 the attribute value
    * @param aHostId2 the attribute value
    * @param aTechContactId the attribute value
    * @return the Wholesale Server Protocol Server response
5   * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_CREATE_DOMAIN
10  * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_ID1
    * @see NSIWholesaleAPI#ATTR_HOST_ID2
15  * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
20  * @see
    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
    *
    */
25  public WSPResponse createDomain(
        String pCustomerId,
        String pCustomerPassword,
        String aDomainName,
        String aHostId1,
        String aHostId2,
        String aTechContactId
30  )
    throws NSIWholesaleException {
        return createDomain(
35  pCustomerId,
        pCustomerPassword,
        aDomainName,
        aHostId1,
        aHostId2,
        aTechContactId,
40  new Properties(),
        0);
    }
    /**
45  * This method registers a new domain product for an existing
    * customer account. This request assumes that the domain
    * name will be hosted by some organization other than
    * Network Solutions.
    * <BR>NOTE: This command is asynchronous. Receipt of the
50  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
55  Expected Result Property Key(s) Returned:
        ATTR_ISP_PASSWORD
    * </PRE><BR>

```

```

    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
    * @param aDomainName the attribute value
    * @param aHostId1 the attribute value
5    * @param aHostId2 the attribute value
    * @param aTechContactId the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
10    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_CREATE_DOMAIN
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
15    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_ID1
    * @see NSIWholesaleAPI#ATTR_HOST_ID2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
-20    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see
25    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
    *
    */
    public WSPResponse createDomain(
30        String pCustomerId,
        String pCustomerPassword,
        String aDomainName,
        String aHostId1,
        String aHostId2,
35        String aTechContactId,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return createDomain(
            pCustomerId,
40            pCustomerPassword,
            aDomainName,
            aHostId1,
            aHostId2,
            aTechContactId,
45            optionalAttributes,
            0);
    }
    protected WSPResponse createDomain(
        String pCustomerId,
50        String pCustomerPassword,
        String aDomainName,
        String aHostId1,
        String aHostId2,
        String aTechContactId,
55        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {

```

```

        String methodName = className + ".createDomain(StringX6,
Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
5 complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

10        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
15            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
            allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
20 pCustomerPassword);
            allParmsAndAttrs.put(ATTR_DOMAIN_NAME, aDomainName);
            allParmsAndAttrs.put(ATTR_HOST_ID1, aHostId1);
            allParmsAndAttrs.put(ATTR_HOST_ID2, aHostId2);
            allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);
25
            returnCode = conn.makeRequest(COMBO_CREATE_DOMAIN,
allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
30        } catch (Exception e) {
            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
35                System.err.println(methodName + "Caught
ResourceException" + re);
            }
            finally {
                return createDomain(
40                    pCustomerAccountId,
                    pCustomerPassword,
                    aDomainName,
                    aHostId1,
                    aHostId2,
45                    aTechContactId,
                    optionalAttributes,
                    count + 1);
            }
        }
50    }

/**
 * This method registers a DNS Hosting product for an existing
 * customer account.
55 * <BR>NOTE: This command is asynchronous. Receipt of the
 * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.

```

```

* <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
5      ATTR_DNS_HOSTING_CREATED
      ATTR_ISP_PASSWORD
* </PRE><BR>
* @param pCustomerId the parameter value
* @param pCustomerPassword the parameter value
10  * @param aDomainName the attribute value
* @return the Wholesale Server Protocol Server response
* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
15  * @see NSIWholesaleAPI#COMMAND_CREATE
    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
    * @see NSIWholesaleAPI#COMBO_CREATE_DNS_HOSTING
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
20  *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
25  * @see NSIWholesaleAPI#ATTR_DNS_HOSTING_CREATED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
    *
*/
public WSPResponse createDnsHosting(
30      String pCustomerId,
      String pCustomerPassword,
      String aDomainName)
throws NSIWholesaleException {
    return createDnsHosting(
35        pCustomerId,
        pCustomerPassword,
        aDomainName,
        new Properties(),
        0);
40  }
/**
    * This method registers a DNS Hosting product for an existing
    * customer account.
    * <BR>NOTE: This command is asynchronous. Receipt of the
45  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
50      Expected Result Property Key(s) Returned:
        ATTR_DNS_HOSTING_CREATED
        ATTR_ISP_PASSWORD
    * </PRE><BR>
    * @param pCustomerId the parameter value
55  * @param pCustomerPassword the parameter value
    * @param aDomainName the attribute value
    * @param optionalAttributes a property list of optional attributes

```

```

    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_CREATE
5    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
    * @see NSIWholesaleAPI#COMBO_CREATE_DNS_HOSTING
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
10   * @see NSIWholesaleAPI#ATTR_DOMAIN_NAME
    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
15   * @see NSIWholesaleAPI#ATTR_DNS_HOSTING_CREATED
    * @see NSIWholesaleAPI#ATTR_ISP_PASSWORD
    *
    */
    public WSPResponse createDnsHosting(
20        String pCustomerId,
        String pCustomerPassword,
        String aDomainName,
        Properties optionalAttributes)
        throws NSIWholesaleException {
25        return createDnsHosting(
            pCustomerId,
            pCustomerPassword,
            aDomainName,
            optionalAttributes,
30            0);
    }

    protected WSPResponse createDnsHosting(
        String pCustomerId,
        String pCustomerPassword,
35        String aDomainName,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".createDnsHosting(String, String,
40        String, Properties, int) ";
        if (count > retry) {
            rc = refundCard(pCustomerId, aDomainName,
                            CreateDnsHostPrice, CreateDnsHostWNRP,
                            optionalAttributes);
45        throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;
50        // Double check to see if the domain really is available (to
minimize charges)
        // with immediate credits
        try {
            System.out.println("***Attempting to verify domain for DNS
55        Hosting.");
            WSPResponse wspResponse = verifyDomainName( aDomainName);
            if (wspResponse.getMajorResultCode() ==

```

```

WSPResponse.COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED) {
    // See if credit card information is available, if so get
transaction
5       System.out.println("*** Begining to do Charge Card
Validation.");
        rc = ChargeCard(pCustomerAccountId, aDomainName,
                        CreateDomainPrice, CreateDomainWNRp,
                        optionalAttributes);
10         if (rc != 0) {
            // Bad credit card. Return bad code.
            throw new NSIWholesaleException(methodName + "Credit card
declined for:" +
                                                aDomainName);
15         }
        } else {
            // Domain name not available. Return bad code.
            throw new NSIWholesaleException(methodName + "Domain name
not available:" +
20         aDomainName);
        }
    }
    catch (NSIWholesaleException e) {
        throw new NSIWholesaleException("Domain name previously
25 taken:" + aDomainName);
    }

    try {
        conn = (NSIWholesaleConnection)pool.checkOut();
30        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

35        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
pCustomerPassword);
        allParmsAndAttrs.put(ATTR_DOMAIN_NAME, aDomainName);
40        returnCode = conn.makeRequest(COMBO_CREATE_DNS_HOSTING,
allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
45    } catch (Exception e) {
        try {
            pool.release(conn);
        }
        catch (ResourceException re) {
50            System.err.println(methodName + "Caught
ResourceException" + re);
        }
        finally {
            return createDnsHosting(
55                pCustomerAccountId,
                pCustomerPassword,
                aDomainName,

```



```

        optionalAttributes,
        count + 1);
    }
}

5      }

    /**
    * This method allows a customer to modify one or more attributes
    * associated with a specific instance of a Domain product
    10    * registered to that customer.
    * This method will also convert a DNS Hosting product domain
    * into a standard domain.
    * <BR>NOTE: This command is asynchronous. Receipt of the
    * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    15    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * </PRE><BR>
    20    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
    * @param pDomainName the parameter value
    * @param aHostName1 the attribute value
    * @param aHostAddr1 the attribute value
    25    * @param aHostName2 the attribute value
    * @param aHostAddr2 the attribute value
    * @param aTechContactId the attribute value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    30    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    35    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_NAME1
    * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
    * @see NSIWholesaleAPI#ATTR_HOST_NAME2
    40    * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#INVALID_VALUE
    45    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#OBJECT_NOT_FOUND
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    50    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    */
    public WSPResponse modifyDomain(
        String pCustomerId,
    55        String pCustomerPassword,
        String pDomainName,
        String aHostName1,

```

```

        String aHostAddr1,
        String aHostName2,
        String aHostAddr2,
        String aTechContactId
5      )
      throws NSIWholesaleException {
        return modifyDomain(
          pCustomerAccountId,
          pCustomerPassword,
10         pDomainName,
          aHostName1,
          aHostAddr1,
          aHostName2,
          aHostAddr2,
15         aTechContactId,
          new Properties(),
          0);
      }
    /**
20     * This method allows a customer to modify one or more attributes
     * associated with a specific instance of a Domain product
     * registered to that customer.
     * This method will also convert a DNS Hosting product domain
     * into a standard domain.
25     * <BR>NOTE: This command is asynchronous. Receipt of the
     * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
     * indicates that the command has been successfully queued.
     * <BR><PRE>
        Expected Major Result Code:
30     COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
        * </PRE><BR>
        * @param pCustomerAccountId the parameter value
        * @param pCustomerPassword the parameter value
        * @param pDomainName the parameter value
35     * @param aHostName1 the attribute value
        * @param aHostAddr1 the attribute value
        * @param aHostName2 the attribute value
        * @param aHostAddr2 the attribute value
        * @param aTechContactId the attribute value
40     * @param optionalAttributes a property list of optional attributes
        * @return the Wholesale Server Protocol Server response
        * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
        unable to complete command
        * @see NSIWholesaleAPI#COMMAND_MODIFY
45     * @see NSIWholesaleAPI#CLASS_DOMAIN
        * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
        * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
        * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
        * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
50     * @see NSIWholesaleAPI#ATTR_HOST_NAME1
        * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
        * @see NSIWholesaleAPI#ATTR_HOST_NAME2
        * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
        * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
55     *
        * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
        * @see WSPResponse#INVALID_VALUE

```

```

    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#OBJECT_NOT_FOUND
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
5   WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    */
10   public WSPResponse modifyDomain(
        String pCustomerId,
        String pCustomerPassword,
        String pDomainName,
        String aHostName1,
        String aHostAddr1,
15   String aHostName2,
        String aHostAddr2,
        String aTechContactId,
        Properties optionalAttributes)
    throws NSIWholesaleException {
20   return modifyDomain(
        pCustomerId,
        pCustomerPassword,
        pDomainName,
        aHostName1,
        aHostAddr1,
25   aHostName2,
        aHostAddr2,
        aTechContactId,
        optionalAttributes,
        0);
30   }

    protected WSPResponse modifyDomain(
        String pCustomerId,
        String pCustomerPassword,
35   String pDomainName,
        String aHostName1,
        String aHostAddr1,
        String aHostName2,
        String aHostAddr2,
        String aTechContactId,
        Properties optionalAttributes,
40   int count)
    throws NSIWholesaleException {
        String methodName = className + ".modifyDomain(StringX8,
45   Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
50   NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
55   conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes

```

```

        propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
5      pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
        pCustomerPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);
        allParmsAndAttrs.put(ATTR_HOST_NAME1, aHostName1);
        allParmsAndAttrs.put(ATTR_HOST_ADDR1, aHostAddr1);
10      allParmsAndAttrs.put(ATTR_HOST_NAME2, aHostName2);
        allParmsAndAttrs.put(ATTR_HOST_ADDR2, aHostAddr2);
        allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);

        returnCode = conn.makeRequest(COMBO_MODIFY_DOMAIN,
15      allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
        try {
20          pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
25        }
        finally {
            return modifyDomain(
                pCustomerAccountId,
                pCustomerPassword,
30              pDomainName,
                aHostName1,
                aHostAddr1,
                aHostName2,
                aHostAddr2,
35              aTechContactId,
                optionalAttributes,
                count + 1);
        }
    }
40  }

/**
 * This method allows a customer to modify one or more attributes
45  * associated with a specific instance of a Domain product
 * registered to that customer.
 * This method will also convert a DNS Hosting product domain
 * into a standard domain.
 * <BR>NOTE: This command is asynchronous. Receipt of the
50  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.
 * <BR><PRE>
    Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
55  * </PRE><BR>
 * @param pCustomerAccountId the parameter value
 * @param pCustomerPassword the parameter value

```

```

    * @param pDomainName the parameter value
    * @param aHostId1 the attribute value
    * @param aHostId2 the attribute value
    * @param aTechContactId the attribute value
5    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
    * @see NSIWholesaleAPI#CLASS_DOMAIN
10    * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_ID1
15    * @see NSIWholesaleAPI#ATTR_HOST_ID2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#INVALID_VALUE
20    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#OBJECT_NOT_FOUND
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
25    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    */
    public WSPResponse modifyDomain(
        String pCustomerId,
30        String pCustomerPassword,
        String pDomainName,
        String aHostId1,
        String aHostId2,
        String aTechContactId
35    )
    throws NSIWholesaleException {
        return modifyDomain(
            pCustomerId,
            pCustomerPassword,
40            pDomainName,
            aHostId1,
            aHostId2,
            aTechContactId,
            new Properties(),
45            0);
    }
    /**
    * This method allows a customer to modify one or more attributes
    * associated with a specific instance of a Domain product
50    * registered to that customer.
    * This method will also convert a DNS Hosting product domain
    * into a standard domain.
    * <BR>NOTE: This command is asynchronous. Receipt of the
    * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
55    * indicates that the command has been successfully queued.
    * <BR><PRE>

```

Expected Major Result Code:

COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED

```

5      * </PRE><BR>
      * @param pCustomerId the parameter value
      * @param pCustomerPassword the parameter value
      * @param pDomainName the parameter value
      * @param aHostId1 the attribute value
      * @param aHostId2 the attribute value
10     * @param aTechContactId the attribute value
      * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
15     * @see NSIWholesaleAPI#COMMAND_MODIFY
      * @see NSIWholesaleAPI#CLASS_DOMAIN
      * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
      * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
      * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
20     * @see NSIWholesaleAPI#ATTR_HOST_ID1
      * @see NSIWholesaleAPI#ATTR_HOST_ID2
      * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
      *
      * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
25     * @see WSPResponse#INVALID_VALUE
      * @see WSPResponse#GENERAL_ACCESS_DENIED
      * @see WSPResponse#OBJECT_NOT_FOUND
      * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      * @see
30     WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
      * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
      *
      */
      public WSPResponse modifyDomain(
35          String pCustomerId,
          String pCustomerPassword,
          String pDomainName,
          String aHostId1,
          String aHostId2,
40          String aTechContactId,
          Properties optionalAttributes)
          throws NSIWholesaleException {
          return modifyDomain(
45              pCustomerId,
              pCustomerPassword,
              pDomainName,
              aHostId1,
              aHostId2,
50              aTechContactId,
              optionalAttributes,
              0);
      }
      protected WSPResponse modifyDomain(
55          String pCustomerId,
          String pCustomerPassword,
          String pDomainName,
          String aHostId1,

```

```

        String aHostId2,
        String aTechContactId,
        Properties optionalAttributes,
        int count)
5      throws NSIWholesaleException {
        String methodName = className + ".modifyDomain(StringX6,
        Properties, int) ";
        if (count > retry) {
10      throw new NSIWholesaleException(methodName + "Unable to
        complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

15      try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
20      propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
        pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
25      pCustomerPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);
        allParmsAndAttrs.put(ATTR_HOST_ID1, aHostId1);
        allParmsAndAttrs.put(ATTR_HOST_ID2, aHostId2);
        allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);
30      returnCode = conn.makeRequest (COMBO_MODIFY_DOMAIN,
        allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
35      } catch (Exception e) {
        try {
            pool.release(conn);
        }
        catch (ResourceException re) {
40      System.err.println(methodName + "Caught
        ResourceException" + re);
        }
        finally {
            return modifyDomain(
45      pCustomerAccountId,
            pCustomerPassword,
            pDomainName,
            aHostId1,
            aHostId2,
50      aTechContactId,
            optionalAttributes,
            count + 1);
        }
    }
55      }

```

```

/**
 * This method allows a technical contact of a customer's domain
 * product to modify one or more attributes
 * associated with that specific instance of the Domain product
5  * registered to that customer.
 * This method will also convert a DNS Hosting product domain
 * into a standard domain.
 * <BR>NOTE: This command is asynchronous. Receipt of the
 * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
10  * indicates that the command has been successfully queued.
 * <BR><PRE>
      Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 * </PRE><BR>
15  * @param pCustomerId the parameter value
 * @param pTechContactId the parameter value
 * @param pTechContactPassword the parameter value
 * @param pDomainName the parameter value
 * @param aHostName1 the attribute value
20  * @param aHostAddr1 the attribute value
 * @param aHostName2 the attribute value
 * @param aHostAddr2 the attribute value
 * @param optionalAttributes a property list of optional attributes
25  * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
 * @see NSIWholesaleAPI#COMMAND_MODIFY
 * @see NSIWholesaleAPI#CLASS_DOMAIN
30  * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
 * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
 * @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID
 * @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD
 * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
35  * @see NSIWholesaleAPI#ATTR_HOST_NAME1
 * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
 * @see NSIWholesaleAPI#ATTR_HOST_NAME2
 * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
 * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
40  *
 * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
 * @see WSPResponse#INVALID_VALUE
 * @see WSPResponse#GENERAL_ACCESS_DENIED
 * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
45  * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
 *
 */
public WSPResponse modifyDomain(
50  String pCustomerId,
    String pTechContactId,
    String pTechContactPassword,
    String pDomainName,
    String aHostName1,
55  String aHostAddr1,
    String aHostName2,
    String aHostAddr2,

```



```

        String aTechContactId
    )
    throws NSIWholesaleException {
        return modifyDomain(
5           pCustomerId,
            pTechContactId,
            pTechContactPassword,
            pDomainName,
            aHostName1,
10          aHostAddr1,
            aHostName2,
            aHostAddr2,
            aTechContactId,
            new Properties(),
15          0);
    }
    /**
    * This method allows a technical contact of a customer's domain
    * product to modify one or more attributes
    * associated with that specific instance of the Domain product
    * registered to that customer.
    * This method will also convert a DNS Hosting product domain
    * into a standard domain.
    * <BR>NOTE: This command is asynchronous. Receipt of the
25  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
30  * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pTechContactId the parameter value
    * @param pTechContactPassword the parameter value
    * @param pDomainName the parameter value
35  * @param aHostName1 the attribute value
    * @param aHostAddr1 the attribute value
    * @param aHostName2 the attribute value
    * @param aHostAddr2 the attribute value
    * @param aTechContactId the attribute value
40  * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
45  * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD
50  * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_NAME1
    * @see NSIWholesaleAPI#ATTR_HOST_ADDR1
    * @see NSIWholesaleAPI#ATTR_HOST_NAME2
    * @see NSIWholesaleAPI#ATTR_HOST_ADDR2
55  * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR

```

```

        * @see WSPResponse#INVALID_VALUE
        * @see WSPResponse#GENERAL_ACCESS_DENIED
        * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
        * @see
5      WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        *
        */
        public WSPResponse modifyDomain(
            String pCustomerId,
10          String pTechContactId,
            String pTechContactPassword,
            String pDomainName,
            String aHostName1,
            String aHostAddr1,
15          String aHostName2,
            String aHostAddr2,
            String aTechContactId,
            Properties optionalAttributes)
        throws NSIWholesaleException {
20          return modifyDomain(
                pCustomerId,
                pTechContactId,
                pTechContactPassword,
                pDomainName,
25          aHostName1,
                aHostAddr1,
                aHostName2,
                aHostAddr2,
                aTechContactId,
                optionalAttributes,
30          0);
        }

        protected WSPResponse modifyDomain(
            String pCustomerId,
35          String pTechContactId,
            String pTechContactPassword,
            String pDomainName,
            String aHostName1,
            String aHostAddr1,
40          String aHostName2,
            String aHostAddr2,
            String aTechContactId,
            Properties optionalAttributes,
            int count)
45          throws NSIWholesaleException {
            String methodName = className + ".modifyDomain(StringX9,
            Properties, int) ";
            if (count > retry) {
                throw new NSIWholesaleException(methodName + "Unable to
50          complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;

55          try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

```

```

        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

5         allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
        pCustomerAccountId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_PASSWORD,
10        pTechContactPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);
        allParmsAndAttrs.put(ATTR_HOST_NAME1, aHostName1);
        allParmsAndAttrs.put(ATTR_HOST_ADDR1, aHostAddr1);
        allParmsAndAttrs.put(ATTR_HOST_NAME2, aHostName2);
        allParmsAndAttrs.put(ATTR_HOST_ADDR2, aHostAddr2);
15        allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);

        returnCode = conn.makeRequest(COMBO_MODIFY_DOMAIN,
        allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
20        } catch (Exception e) {
            try {
                pool.release(conn);
            }
25            catch (ResourceException re) {
                System.err.println(methodName + "Caught
ResourceException" + re);
            }
            finally {
30                return modifyDomain(
                    pCustomerAccountId,
                    pTechContactId,
                    pTechContactPassword,
                    pDomainName,
35                    aHostName1,
                    aHostAddr1,
                    aHostName2,
                    aHostAddr2,
                    aTechContactId,
40                    optionalAttributes,
                    count + 1);
            }
        }
45    }

/**
 * This method allows a technical contact of a customer's domain
 * product to modify one or more attributes
50 * associated with that specific instance of the Domain product
 * registered to that customer.
 * This method will also convert a DNS Hosting product domain
 * into a standard domain.
 * <BR>NOTE: This command is asynchronous. Receipt of the
55 * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.
 * <BR><PRE>

```

Expected Major Result Code:

COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED

* </PRE>

* @param pCustomerId the parameter value

* @param pTechContactId the parameter value

* @param pTechContactPassword the parameter value

* @param pDomainName the parameter value

* @param aHostId1 the attribute value

* @param aHostId2 the attribute value

* @param aTechContactId the attribute value

* @return the Wholesale Server Protocol Server response

* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command

* @see NSIWholesaleAPI#COMMAND_MODIFY

* @see NSIWholesaleAPI#CLASS_DOMAIN

* @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN

* @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID

* @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID

* @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD

* @see NSIWholesaleAPI#PARM_DOMAIN_NAME

* @see NSIWholesaleAPI#ATTR_HOST_ID1

* @see NSIWholesaleAPI#ATTR_HOST_ID2

* @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID

*

* @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR

* @see WSPResponse#INVALID_VALUE

* @see WSPResponse#GENERAL_ACCESS_DENIED

* @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED

* @see

WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED

*

*/

public WSPResponse modifyDomain(
 String pCustomerId,
 String pTechContactId,
 String pTechContactPassword,
 String pDomainName,
 String aHostId1,
 String aHostId2,
 String aTechContactId
)

throws NSIWholesaleException {

return modifyDomain(
 pCustomerId,
 pTechContactId,
 pTechContactPassword,
 pDomainName,
 aHostId1,
 aHostId2,
 aTechContactId,
 new Properties(),
 0);
 }

/**

* This method allows a technical contact of a customer's domain

* product to modify one or more attributes

* associated with that specific instance of the Domain product

```

    * registered to that customer.
    * This method will also convert a DNS Hosting product domain
    * into a standard domain.
    * <BR>NOTE: This command is asynchronous. Receipt of the
5    * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
    * indicates that the command has been successfully queued.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
10    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pTechContactId the parameter value
    * @param pTechContactPassword the parameter value
    * @param pDomainName the parameter value
15    * @param aHostId1 the attribute value
    * @param aHostId2 the attribute value
    * @param aTechContactId the attribute value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
20    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
25    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD
    * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
    * @see NSIWholesaleAPI#ATTR_HOST_ID1
30    * @see NSIWholesaleAPI#ATTR_HOST_ID2
    * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
    *
    * @see WSPResponse#COMMAND_FAILED_UNKNOWN_ERROR
    * @see WSPResponse#INVALID_VALUE
35    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *
40    */
    public WSPResponse modifyDomain(
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
45        String pDomainName,
        String aHostId1,
        String aHostId2,
        String aTechContactId,
        Properties optionalAttributes)
50    throws NSIWholesaleException {
        return modifyDomain(
            pCustomerId,
            pTechContactId,
            pTechContactPassword,
55            pDomainName,
            aHostId1,
            aHostId2,

```

```

        aTechContactId,
        optionalAttributes,
        0);
    }
5    protected WSPResponse modifyDomain(
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
        String pDomainName,
10    String aHostId1,
        String aHostId2,
        String aTechContactId,
        Properties optionalAttributes,
        int count)
15    throws NSIWholesaleException {
        String methodName = className + ".modifyDomain(StringX7,
        Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
20    complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

25    try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
30    propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
        pCustomerId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);
35    allParmsAndAttrs.put(PARM_TECH_CONTACT_PASSWORD,
        pTechContactPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);
        allParmsAndAttrs.put(ATTR_HOST_ID1, aHostId1);
        allParmsAndAttrs.put(ATTR_HOST_ID2, aHostId2);
40    allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);

        returnCode = conn.makeRequest(COMBO_MODIFY_DOMAIN,
        allParmsAndAttrs);
        pool.checkIn(conn);
45    return returnCode;
    } catch (Exception e) {
        try {
            pool.release(conn);
        }
50    catch (ResourceException re) {
        System.err.println(methodName + "Caught
        ResourceException" + re);
    }
    finally {
55    return modifyDomain(
        pCustomerId,
        pTechContactId,

```

```

        pTechContactPassword,
        pDomainName,
        aHostId1,
        aHostId2,
5         aTechContactId,
        optionalAttributes,
        count + 1);
    }
}
10 }

/**
 * This method allows an Internet Service Provider (ISP)
15 * to modify the Technical Contact Id of a specific domain product.
 * <BR>NOTE: This command is asynchronous. Receipt of the
 * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.
 * <BR><PRE>
20     Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 * </PRE><BR>
 * @param pCustomerId the parameter value
 * @param pIspPassword the parameter value
25 * @param pDomainName the parameter value
 * @param aTechContactId the attribute value
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
30 * @see NSIWholesaleAPI#COMMAND_MODIFY
 * @see NSIWholesaleAPI#CLASS_DOMAIN
 * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
 * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
 * @see NSIWholesaleAPI#PARM_ISP_PASSWORD
35 * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
 * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
 *
 * @see WSPResponse#INVALID_VALUE
 * @see WSPResponse#GENERAL_ACCESS_DENIED
40 * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 */
public WSPResponse modifyDomain(
45     String pCustomerId,
     String pIspPassword,
     String pDomainName,
     String aTechContactId
)
throws NSIWholesaleException {
50     return modifyDomain(
        pCustomerId,
        pIspPassword,
        pDomainName,
        aTechContactId,
55     new Properties(),
        0);
}

```

```

/**
 * This method allows an Internet Service Provider (ISP)
 * to modify the Technical Contact Id of a specific domain product.
 * <BR>NOTE: This command is asynchronous. Receipt of the
5  * COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED code
 * indicates that the command has been successfully queued.
 * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
10  * </PRE><BR>
 * @param pCustomerId the parameter value
 * @param pIspPassword the parameter value
 * @param pDomainName the parameter value
 * @param aTechContactId the attribute value
15  * @param optionalAttributes a property list of optional attributes
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
 * @see NSIWholesaleAPI#COMMAND_MODIFY
20  * @see NSIWholesaleAPI#CLASS_DOMAIN
 * @see NSIWholesaleAPI#COMBO_MODIFY_DOMAIN
 * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
 * @see NSIWholesaleAPI#PARAM_ISP_PASSWORD
 * @see NSIWholesaleAPI#PARAM_DOMAIN_NAME
25  * @see NSIWholesaleAPI#ATTR_TECH_CONTACT_ID
 *
 * @see WSPResponse#INVALID_VALUE
 * @see WSPResponse#GENERAL_ACCESS_DENIED
 * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
30  *
 */
public WSPResponse modifyDomain(
    String pCustomerId,
    String pIspPassword,
35  String pDomainName,
    String aTechContactId,
    Properties optionalAttributes)
throws NSIWholesaleException {
    return modifyDomain(
40  pCustomerId,
    pIspPassword,
    pDomainName,
    aTechContactId,
    optionalAttributes,
45  0);
}

protected WSPResponse modifyDomain(
    String pCustomerId,
    String pIspPassword,
50  String pDomainName,
    String aTechContactId,
    Properties optionalAttributes,
    int count)
throws NSIWholesaleException {
55  String methodName = className + ".modifyDomain(StringX4,
    Properties, int) ";
    if (count > retry) {

```



```

        throw new NSIWholesaleException(methodName + "Unable to
complete command");
    }
    NSIWholesaleConnection conn = null;
5    WSPResponse returnCode;

    try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();
10        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
15    pCustomerId);
        allParmsAndAttrs.put(PARM_ISP_PASSWORD, pIspPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);
        allParmsAndAttrs.put(ATTR_TECH_CONTACT_ID, aTechContactId);

        returnCode = conn.makeRequest(COMBO_MODIFY_DOMAIN,
20    allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
25        try {
            pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
30    ResourceException" + re);
        }
        finally {
            return modifyDomain(
                pCustomerId,
35                pIspPassword,
                pDomainName,
                aTechContactId,
                optionalAttributes,
                count + 1);
40        }
    }
}

45    /**
     * This method is used to modify one or more attributes
     * of an existing individual customer account.
     * <BR><PRE>
        Expected Major Result Code:
50    COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
     * </PRE><BR>
     * @param pCustomerId the parameter value
     * @param pCustomerPassword the parameter value
     * @return the Wholesale Server Protocol Server response
55    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
     * @see NSIWholesaleAPI#COMMAND_MODIFY

```

```

5      * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
      * @see NSIWholesaleAPI#COMBO_MODIFY_INDIVIDUAL_ACCOUNT
      * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
      * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
      *
      * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#ATTR_FIRST_NAME
      * @see NSIWholesaleAPI#ATTR_LAST_NAME
10     * @see NSIWholesaleAPI#ATTR_ADDRESS1
      * @see NSIWholesaleAPI#ATTR_ADDRESS2
      * @see NSIWholesaleAPI#ATTR_ADDRESS3
      * @see NSIWholesaleAPI#ATTR_ADDRESS4
      * @see NSIWholesaleAPI#ATTR_ADDRESS5
      * @see NSIWholesaleAPI#ATTR_CITY
15     * @see NSIWholesaleAPI#ATTR_STATE
      * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
      * @see NSIWholesaleAPI#ATTR_PHONE
      * @see NSIWholesaleAPI#ATTR_FAX
20     * @see NSIWholesaleAPI#ATTR_EMAIL
      * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
      * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
      *
25     * @see WSPResponse#GENERAL_ACCESS_DENIED
      * @see WSPResponse#INVALID_VALUE
      * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      *
      */
      public WSPResponse modifyIndividualAccount(
30         String pCustomerId,
          String pCustomerPassword)
          throws NSIWholesaleException {
          return modifyIndividualAccount(
35              pCustomerId,
              pCustomerPassword,
              new Properties(),
              0);
      }
      /**
40     * This method is used to modify one or more attributes
      * of an existing individual customer account.
      * <BR><PRE>
          Optional Attribute Key(s):
45             ATTR_CUSTOMER_PASSWORD
              ATTR_FIRST_NAME
              ATTR_LAST_NAME
              ATTR_ADDRESS1
              ATTR_ADDRESS2
              ATTR_ADDRESS3
50             ATTR_ADDRESS4
              ATTR_ADDRESS5
              ATTR_CITY
              ATTR_STATE
              ATTR_POSTAL_CODE
55             ATTR_COUNTRY_CODE
              ATTR_PHONE
              ATTR_FAX

```

ATTR_EMAIL
ATTR_AUTH_QUESTION
ATTR_AUTH_ANSWER

```

5      Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
* </PRE><BR>
* @param pCustomerId the parameter value
* @param pCustomerPassword the parameter value
10  * @param optionalAttributes a property list of optional attributes
* @return the Wholesale Server Protocol Server response
* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
15  * @see NSIWholesaleAPI#COMMAND_MODIFY
* @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
* @see NSIWholesaleAPI#COMBO_MODIFY_INDIVIDUAL_ACCOUNT
* @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
* @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
*
20  * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
* @see NSIWholesaleAPI#ATTR_FIRST_NAME
* @see NSIWholesaleAPI#ATTR_LAST_NAME
* @see NSIWholesaleAPI#ATTR_ADDRESS1
* @see NSIWholesaleAPI#ATTR_ADDRESS2
25  * @see NSIWholesaleAPI#ATTR_ADDRESS3
* @see NSIWholesaleAPI#ATTR_ADDRESS4
* @see NSIWholesaleAPI#ATTR_ADDRESS5
* @see NSIWholesaleAPI#ATTR_CITY
* @see NSIWholesaleAPI#ATTR_STATE
30  * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
* @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
* @see NSIWholesaleAPI#ATTR_PHONE
* @see NSIWholesaleAPI#ATTR_FAX
* @see NSIWholesaleAPI#ATTR_EMAIL
35  * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
* @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
*
* @see WSPResponse#GENERAL_ACCESS_DENIED
* @see WSPResponse#INVALID_VALUE
40  * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
*
*/
public WSPResponse modifyIndividualAccount(
45      String pCustomerId,
      String pCustomerPassword,
      Properties optionalAttributes)
throws NSIWholesaleException {
    return modifyIndividualAccount(
50      pCustomerId,
      pCustomerPassword,
      optionalAttributes,
      0);
}
protected WSPResponse modifyIndividualAccount(
55      String pCustomerId,
      String pCustomerPassword,
      Properties optionalAttributes,

```

```

        int count)
        throws NSIWholesaleException {
            String methodName = className + ".modifyIndividualAccount(String,
String, Properties, int) ";
5           if (count > retry) {
                throw new NSIWholesaleException(methodName + "Unable to
complete command");
            }
            NSIWholesaleConnection conn = null;
10           WSPResponse returnCode;

            try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();
15                // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);

                allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
20                pCustomerAccountId);
                allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
                pCustomerPassword);

                returnCode =
25                conn.makeRequest(COMBO_MODIFY_INDIVIDUAL_ACCOUNT, allParmsAndAttrs);
                pool.checkIn(conn);
                return returnCode;
            } catch (Exception e) {
                try {
30                    pool.release(conn);
                }
                catch (ResourceException re) {
                    System.err.println(methodName + "Caught
ResourceException" + re);
35                }
                finally {
                    return modifyIndividualAccount(
                        pCustomerAccountId,
                        pCustomerPassword,
40                        optionalAttributes,
                        count + 1);
                }
            }
45        }

/**
 * This method is used to modify one or more attributes
 * of an existing business (organization) customer account.
50 * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
        * </PRE><BR>
        * @param pCustomerAccountId the parameter value
        * @param pCustomerPassword the parameter value
55 * @return the Wholesale Server Protocol Server response

```

```

    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
5    * @see NSIWholesaleAPI#COMBO_MODIFY_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    *
10    * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
    * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
    * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
    * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_ADDRESS2
15    * @see NSIWholesaleAPI#ATTR_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
20    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_FAX
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
25    * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
30    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
35    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
40    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    *
45    */
    public WSPResponse modifyBusinessAccount(
        String pCustomerId,
        String pCustomerPassword)
        throws NSIWholesaleException {
50        return modifyBusinessAccount(
            pCustomerId,
            pCustomerPassword,
            new Properties(),
            0);
55    }
    /**
    * This method is used to modify one or more attributes

```

* of an existing business (organization) customer account.
 *
<PRE>

Optional Attribute Key(s):

5 ATTR_CUSTOMER_PASSWORD
 ATTR_COMPANY_NAME
 ATTR_COMPANY_TYPE
 ATTR_ADDRESS1
 ATTR_ADDRESS2
 10 ATTR_ADDRESS3
 ATTR_ADDRESS4
 ATTR_ADDRESS5
 ATTR_CITY
 ATTR_STATE
 ATTR_POSTAL_CODE
 15 ATTR_COUNTRY_CODE
 ATTR_PHONE
 ATTR_FAX
 ATTR_EMAIL
 ATTR_AUTH_QUESTION
 20 ATTR_AUTH_ANSWER
 ATTR_LEGAL_CONTACT_FIRST_NAME
 ATTR_LEGAL_CONTACT_LAST_NAME
 ATTR_LEGAL_CONTACT_ADDRESS1
 ATTR_LEGAL_CONTACT_ADDRESS2
 25 ATTR_LEGAL_CONTACT_ADDRESS3
 ATTR_LEGAL_CONTACT_ADDRESS4
 ATTR_LEGAL_CONTACT_ADDRESS5
 ATTR_LEGAL_CONTACT_CITY
 ATTR_LEGAL_CONTACT_STATE
 30 ATTR_LEGAL_CONTACT_POSTAL_CODE
 ATTR_LEGAL_CONTACT_COUNTRY_CODE
 ATTR_LEGAL_CONTACT_PHONE
 ATTR_LEGAL_CONTACT_FAX
 ATTR_LEGAL_CONTACT_EMAIL

Expected Major Result Code:

COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED

* </PRE>

 * @param pCustomerId the parameter value
 40 * @param pCustomerPassword the parameter value
 * @param optionalAttributes a property list of optional attributes
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
 unable to complete command
 45 * @see NSIWholesaleAPI#COMMAND_MODIFY
 * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
 * @see NSIWholesaleAPI#COMBO_MODIFY_BUSINESS_ACCOUNT
 * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
 * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
 50 *
 * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
 * @see NSIWholesaleAPI#ATTR_COMPANY_NAME
 * @see NSIWholesaleAPI#ATTR_COMPANY_TYPE
 * @see NSIWholesaleAPI#ATTR_ADDRESS1
 55 * @see NSIWholesaleAPI#ATTR_ADDRESS2
 * @see NSIWholesaleAPI#ATTR_ADDRESS3
 * @see NSIWholesaleAPI#ATTR_ADDRESS4

```

    * @see NSIWholesaleAPI#ATTR_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
5    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_FAX
    * @see NSIWholesaleAPI#ATTR_EMAIL
    * @see NSIWholesaleAPI#ATTR_AUTH_QUESTION
10   * @see NSIWholesaleAPI#ATTR_AUTH_ANSWER
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS2
15   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS3
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS4
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_ADDRESS5
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_CITY
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_STATE
20   * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_COUNTRY_CODE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_PHONE
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
    * @see NSIWholesaleAPI#ATTR_LEGAL_CONTACT_FAX
25   *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#INVALID_VALUE
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    *
30   */
    public WSPResponse modifyBusinessAccount(
        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes)
35   throws NSIWholesaleException {
        return modifyBusinessAccount(
            pCustomerId,
            pCustomerPassword,
            optionalAttributes,
40            0);
    }

    protected WSPResponse modifyBusinessAccount(
        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes,
45        int count)
        throws NSIWholesaleException {
        String methodName = className + ".modifyBusinessAccount(String,
String, Properties, int) ";
50        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
55        WSPResponse returnCode;

        try {

```

```

        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
5      propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
10     pCustomerPassword);

        returnCode = conn.makeRequest(COMBO_MODIFY_BUSINESS_ACCOUNT,
allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
15     } catch (Exception e) {
        try {
            pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
        }
        finally {
25             return modifyBusinessAccount(
                    pCustomerAccountId,
                    pCustomerPassword,
                    optionalAttributes,
                    count + 1);
        }
30     }
    }

    /**
35     * This method is used to modify one or more attributes
    * of an existing technical contact.
    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
40     * </PRE><BR>
    * @param pTechContactId the parameter value
    * @param pTechContactPassword the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
45     unable to complete command
    * @see NSIWholesaleAPI#COMMAND_MODIFY
    * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#COMBO_MODIFY_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID
50     * @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD
    *
    * @see NSIWholesaleAPI#ATTR_FIRST_NAME
    * @see NSIWholesaleAPI#ATTR_LAST_NAME
    * @see NSIWholesaleAPI#ATTR_ORGANIZATION
55     * @see NSIWholesaleAPI#ATTR_ADDRESS1
    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE

```



```

5      * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
      * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
      * @see NSIWholesaleAPI#ATTR_PHONE
      * @see NSIWholesaleAPI#ATTR_FAX
      * @see NSIWholesaleAPI#ATTR_EMAIL
      *
      * @see WSPResponse#GENERAL_ACCESS_DENIED
      * @see WSPResponse#INVALID_VALUE
10     * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      *
      */
      public WSPResponse modifyTechnicalContact(
          String pTechContactId,
          String pTechContactPassword)
15     throws NSIWholesaleException {
          return modifyTechnicalContact(
              pTechContactId,
              pTechContactPassword,
20              new Properties(),
              0);
      }
      /**
      * This method is used to modify one or more attributes
      * of an existing technical contact.
25     * <BR><PRE>
          Optional Attribute Key(s):
              ATTR_FIRST_NAME
              ATTR_LAST_NAME
              ATTR_ORGANIZATION
30             ATTR_ADDRESS1
              ATTR_CITY
              ATTR_STATE
              ATTR_POSTAL_CODE
              ATTR_COUNTRY_CODE
              ATTR_EMAIL
35             ATTR_FAX
              ATTR_PHONE

          Expected Major Result Code:
40     COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      * </PRE><BR>
      * @param pTechContactId the parameter value
      * @param pTechContactPassword the parameter value
      * @param optionalAttributes a property list of optional attributes
45     * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
      unable to complete command
      * @see NSIWholesaleAPI#COMMAND_MODIFY
      * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
50     * @see NSIWholesaleAPI#COMBO_MODIFY_TECHNICAL_CONTACT
      * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
      * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_PASSWORD
      *
      * @see NSIWholesaleAPI#ATTR_FIRST_NAME
55     * @see NSIWholesaleAPI#ATTR_LAST_NAME
      * @see NSIWholesaleAPI#ATTR_ORGANIZATION
      * @see NSIWholesaleAPI#ATTR_ADDRESS1

```

```

    * @see NSIWholesaleAPI#ATTR_CITY
    * @see NSIWholesaleAPI#ATTR_STATE
    * @see NSIWholesaleAPI#ATTR_POSTAL_CODE
    * @see NSIWholesaleAPI#ATTR_COUNTRY_CODE
5    * @see NSIWholesaleAPI#ATTR_PHONE
    * @see NSIWholesaleAPI#ATTR_FAX
    * @see NSIWholesaleAPI#ATTR_EMAIL
    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
10    * @see WSPResponse#INVALID_VALUE
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    *
    */
    public WSPResponse modifyTechnicalContact(
15        String pTechContactId,
        String pTechContactPassword,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return modifyTechnicalContact(
20            pTechContactId,
            pTechContactPassword,
            optionalAttributes,
            0);
    }
25    protected WSPResponse modifyTechnicalContact(
        String pTechContactId,
        String pTechContactPassword,
        Properties optionalAttributes,
        int count)
30        throws NSIWholesaleException {
        String methodName = className + ".modifyTechnicalContact(String,
String, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
35 complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

40        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
45            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);
            allParmsAndAttrs.put(PARM_TECH_CONTACT_PASSWORD,
pTechContactPassword);
50            returnCode =
conn.makeRequest(COMBO_MODIFY_TECHNICAL_CONTACT, allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
55        } catch (Exception e) {
            try {
                pool.release(conn);
            }
        }
    }

```

```

    }
    catch (ResourceException re) {
        System.err.println(methodName + "Caught
ResourceException" + re);
5      }
      finally {
          return modifyTechnicalContact(
              pTechContactId,
              pTechContactPassword,
10             optionalAttributes,
              count + 1);
          }
      }
15  }

/**
 * This method is used to modify/reset a customer's account password.
 * <BR><PRE>
20      Expected Major Result Code:
      COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      * </PRE><BR>
      * @param pCustomerId the parameter value
      * @param aCustomerPassword the attribute value
25      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
      unable to complete command
      * @see NSIWholesaleAPI#COMMAND_MODIFY
      * @see NSIWholesaleAPI#CLASS_PASSWORD
30      * @see NSIWholesaleAPI#COMBO_MODIFY_PASSWORD
      * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
      * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
      *
      * @see WSPResponse#GENERAL_ACCESS_DENIED
35      * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      *
      */
    public WSPResponse modifyPassword(
        String pCustomerId,
40        String aCustomerPassword)
        throws NSIWholesaleException {
        return modifyPassword(
            pCustomerId,
            aCustomerPassword,
45            new Properties(),
            0);
    }
    /**
 * This method is used to modify/reset a customer's account password.
 * <BR><PRE>
50      Expected Major Result Code:
      COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      * </PRE><BR>
      * @param pCustomerId the parameter value
      * @param aCustomerPassword the attribute value
55      * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response

```

```

        * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
        * @see NSIWholesaleAPI#COMMAND_MODIFY
        * @see NSIWholesaleAPI#CLASS_PASSWORD
5      * @see NSIWholesaleAPI#COMBO_MODIFY_PASSWORD
        * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
        * @see NSIWholesaleAPI#ATTR_CUSTOMER_PASSWORD
        *
        * @see WSPResponse#GENERAL_ACCESS_DENIED
10     * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
        *
    */
    public WSPResponse modifyPassword(
        String pCustomerId,
15     String aCustomerPassword,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return modifyPassword(
            pCustomerId,
20     aCustomerPassword,
            optionalAttributes,
            0);
    }

    protected WSPResponse modifyPassword(
25     String pCustomerId,
        String aCustomerPassword,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
30     String methodName = className + ".modifyPassword(String, String,
        Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
35     }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
40     conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);
45     allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
        pCustomerId);
        allParmsAndAttrs.put(ATTR_CUSTOMER_PASSWORD,
        aCustomerPassword);
50     returnCode = conn.makeRequest(COMBO_MODIFY_PASSWORD,
        allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
55     } catch (Exception e) {
        try {
            pool.release(conn);

```

```

    }
    catch (ResourceException re) {
        System.err.println(methodName + "Caught
ResourceException" + re);
5        }
        finally {
            return modifyPassword(
                pCustomerAccountId,
                aCustomerPassword,
10                optionalAttributes,
                count + 1);
        }
    }
}

15 /**
 * This method is used to generate a list of available
 * domain names based on a specified list of keywords.
 * <BR><PRE>
20     Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
        DomainName0:
        DomainName1:
25        DomainName2:
        DomainName3:
        ...etc...
    * </PRE><BR>
    * @param pKeywords the parameter value
30    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_GENERATE
    * @see NSIWholesaleAPI#CLASS_DOMAIN_NAME
35    * @see NSIWholesaleAPI#COMBO_GENERATE_DOMAIN_NAME
    * @see NSIWholesaleAPI#PARM_KEY_WORDS
    *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
40    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    */
    public WSPResponse generateDomainName(
        String pKeywords)
45    throws NSIWholesaleException {
        return generateDomainName(
            pKeywords,
            new Properties(),
            0);
50    }
}

/**
 * This method is used to generate a list of available
 * domain names based on a specified list of keywords.
 * <BR><PRE>
55     Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:

```

```

DomainName0:
DomainName1:
DomainName2:
DomainName3:
...etc...
5      * </PRE><BR>
      * @param pKeyWords the parameter value
      * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response
10     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
      * @see NSIWholesaleAPI#COMMAND_GENERATE
      * @see NSIWholesaleAPI#CLASS_DOMAIN_NAME
      * @see NSIWholesaleAPI#COMBO_GENERATE_DOMAIN_NAME
15     * @see NSIWholesaleAPI#PARM_KEY_WORDS
      *
      * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
      * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
20     *
    */
    public WSPResponse generateDomainName(
        String pKeyWords,
        Properties optionalAttributes)
25     throws NSIWholesaleException {
        return generateDomainName(
            pKeyWords,
            optionalAttributes,
            0);
30     }
    protected WSPResponse generateDomainName(
        String pKeyWords,
        Properties optionalAttributes,
        int count)
35     throws NSIWholesaleException {
        String methodName = className + ".generateDomainName(String,
Properties, int) ";
        if (count > retry) {
40     complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

45     try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
50     propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_KEY_WORDS, pKeyWords);

        returnCode = conn.makeRequest(COMBO_GENERATE_DOMAIN_NAME,
55     allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;

```

```

        } catch (Exception e) {
            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
ResourceException" + re);
            }
            finally {
                return generateDomainName(
                    pKeywords,
                    optionalAttributes,
                    count + 1);
            }
        }
    }

/**
 * Retrieves a list of all domain products owned by customer.
 * <BR><PRE>
    Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
25         DomainName1:
            DomainName2:
            DomainName3:
            ...etc...
 * </PRE><BR>
30 * @param pCustomerId the parameter value
 * @param pCustomerPassword the parameter value
 * @return the Wholesale Server Protocol Server response
 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
35 * @see NSIWholesaleAPI#COMMAND_LOOKUP
 * @see NSIWholesaleAPI#CLASS_DOMAIN
 * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
 * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
 * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
40 *
 * @see WSPResponse#GENERAL_ACCESS_DENIED
 * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
 * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
45 *
 */
public WSPResponse lookupDomain(
    String pCustomerId,
    String pCustomerPassword)
50 throws NSIWholesaleException {
    return lookupDomain(
        pCustomerId,
        pCustomerPassword,
        new Properties(),
        0);
55 }
/**

```

```

    * Retrieves a list of all domain products owned by customer.
    * <BR><PRE>
      Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
5      Expected Result Property Key(s) Returned:
        DomainName1:
        DomainName2:
        DomainName3:
        ...etc...
10    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
15    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
20    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
25    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    */
    public WSPResponse lookupDomain(
30        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return lookupDomain(
35            pCustomerId,
            pCustomerPassword,
            optionalAttributes,
            0);
    }
    protected WSPResponse lookupDomain(
40        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes,
        int count)
45        throws NSIWholesaleException {
        String methodName = className + ".lookupDomain(String, String,
Properties, int) ";
        if (count > retry) {
50            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

55        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

```



```

        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

5         allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
pCustomerPassword);

10         returnCode = conn.makeRequest(COMBO_LOOKUP_DOMAIN,
allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
15         try {
            pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
20 ResourceException" + re);
        }
        finally {
            return lookupDomain(
25                pCustomerAccountId,
                pCustomerPassword,
                optionalAttributes,
                count + 1);
        }
    }
30 }

/**
 * Retrieves detail information about a specific domain product
35 * registered to a customer. This form of the lookup domain
 * command will also retrieve the status of the domain while
 * it is being processed or after it has failed.
 * <BR><PRE>
    Expected Major Result Code:
40 COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
        ATTR_DOMAIN_NAME
        ATTR_TECH_CONTACT_ID
        ATTR_HOST_NAME1
        ATTR_HOST_ADDR1
45        ATTR_HOST_ID1
        ATTR_HOST_NAME2
        ATTR_HOST_ADDR2
        ATTR_HOST_ID2
50 * </PRE><BR>
 * @param pCustomerAccountId the parameter value
 * @param pCustomerPassword the parameter value
 * @param pDomainName the parameter value
 * @return the Wholesale Server Protocol Server response
55 * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
 * @see NSIWholesaleAPI#COMMAND_LOOKUP

```

```

5      * @see NSIWholesaleAPI#CLASS_DOMAIN
      * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
      * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
      * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
      *
      * @see WSPResponse#GENERAL_ACCESS_DENIED
      * @see WSPResponse#OBJECT_NOT_FOUND
      * @see
10  WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
      * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
      *
      */
      public WSPResponse lookupDomain(
15          String pCustomerId,
          String pCustomerPassword,
          String pDomainName)
      throws NSIWholesaleException {
          return lookupDomain(
20              pCustomerId,
              pCustomerPassword,
              pDomainName,
              new Properties(),
              0);
25      }
      /**
      * Retrieves detail information about a specific domain product
      * registered to a customer. This form of the lookup domain
      * command will also retrieve the status of the domain while
30      * it is being processed or after it has failed.
      * <BR><PRE>
          Expected Major Result Code:
      COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
          Expected Result Property Key(s) Returned:
35              ATTR_DOMAIN_NAME
              ATTR_TECH_CONTACT_ID
              ATTR_HOST_NAME1
              ATTR_HOST_ADDR1
              ATTR_HOST_ID1
40              ATTR_HOST_NAME2
              ATTR_HOST_ADDR2
              ATTR_HOST_ID2
      * </PRE><BR>
      * @param pCustomerId the parameter value
45      * @param pCustomerPassword the parameter value
      * @param pDomainName the parameter value
      * @param optionalAttributes a property list of optional attributes
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
50      unable to complete command
      * @see NSIWholesaleAPI#COMMAND_LOOKUP
      * @see NSIWholesaleAPI#CLASS_DOMAIN
      * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
      * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
55      * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
      * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
      *

```

```

        * @see WSPResponse#GENERAL_ACCESS_DENIED
        * @see WSPResponse#OBJECT_NOT_FOUND
        * @see
5      WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
        */
    public WSPResponse lookupDomain(
10        String pCustomerId,
        String pCustomerPassword,
        String pDomainName,
        Properties optionalAttributes)
        throws NSIWholesaleException {
15        return lookupDomain(
            pCustomerId,
            pCustomerPassword,
            pDomainName,
            optionalAttributes,
            0);
20    }
    protected WSPResponse lookupDomain(
        String pCustomerId,
        String pCustomerPassword,
        String pDomainName,
25        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className + ".lookupDomain(String, String,
30        String, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
            complete command");
        }
        NSIWholesaleConnection conn = null;
35        WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();
40            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
45            pCustomerId);
            allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
            pCustomerPassword);
            allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);

            returnCode = conn.makeRequest(COMBO_LOOKUP_DOMAIN,
50            allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
55            try {
                pool.release(conn);
            }

```

```

        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
        }
5         finally {
            return lookupDomain(
                pCustomerAccountId,
                pCustomerPassword,
                pDomainName,
10             optionalAttributes,
                count + 1);
        }
    }
15
}

/**
 * Retrieves a list of the customer's domain products
 * for which the technical contact id is the technical contact.
20 * <BR><PRE>
    Expected Major Result Code:
    COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
        DomainName1:
        DomainName2:
25        DomainName3:
        ...etc...
    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pTechContactId the parameter value
    * @param pTechContactPassword the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
35 * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
40 * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_PASSWORD
    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
45 WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    */
    public WSPResponse lookupDomainByTechContact(
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword)
    throws NSIWholesaleException {
        return lookupDomainByTechContact(
            pCustomerId,
            pTechContactId,
            pTechContactPassword,
55            new Properties(),
        );
    }
}

```

```

        0);
    }
    /**
     * Retrieves a list of the customer's domain products
     * for which the technical contact id is the technical contact.
     * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
        DomainName1:
        DomainName2:
        DomainName3:
        ...etc...

     * </PRE><BR>
     * @param pCustomerId the parameter value
     * @param pTechContactId the parameter value
     * @param pTechContactPassword the parameter value
     * @param optionalAttributes a property list of optional attributes
     * @return the Wholesale Server Protocol Server response
     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
     * @see NSIWholesaleAPI#COMMAND_LOOKUP
     * @see NSIWholesaleAPI#CLASS_DOMAIN
     * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
     * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
     * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
     * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_PASSWORD
     *
     * @see WSPResponse#GENERAL_ACCESS_DENIED
     * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
     * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
     */
    public WSPResponse lookupDomainByTechContact(
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return lookupDomainByTechContact(
            pCustomerId,
            pTechContactId,
            pTechContactPassword,
            optionalAttributes,
            0);
    }
    protected WSPResponse lookupDomainByTechContact(
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className +
        ".lookupDomainByTechContact(String, String, String, Properties, int) ";
        if (count > retry) {

```

```

        throw new NSIWholesaleException(methodName + "Unable to
complete command");
    }
    NSIWholesaleConnection conn = null;
5    WSPResponse returnCode;

    try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();
10
        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
15    pCustomerAccountId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_PASSWORD,
        pTechContactPassword);

20    returnCode = conn.makeRequest(COMBO_LOOKUP_DOMAIN,
        allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
25    try {
        pool.release(conn);
    }
    catch (ResourceException re) {
        System.err.println(methodName + "Caught
30    ResourceException" + re);
    }
    finally {
        return lookupDomainByTechContact(
35    pCustomerAccountId,
        pTechContactId,
        pTechContactPassword,
        optionalAttributes,
        count + 1);
    }
40    }
}

/**
45    * Retrieves details of a specific domain product owned by customer,
    * for which the technical contact id is the technical contact for
    * the specific product.
    * <BR><PRE>
        Expected Major Result Code:
50    COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_DOMAIN_NAME
            ATTR_TECH_CONTACT_ID
            ATTR_HOST_NAME1
55    ATTR_HOST_ADDR1
            ATTR_HOST_ID1
            ATTR_HOST_NAME2

```

```

ATTR_HOST_ADDR2
ATTR_HOST_ID2
* </PRE><BR>
* @param pCustomerId the parameter value
5  * @param pTechContactId the parameter value
* @param pTechContactPassword the parameter value
* @param pDomainName the parameter value
* @return the Wholesale Server Protocol Server response
* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
10 unable to complete command
* @see NSIWholesaleAPI#COMMAND_LOOKUP
* @see NSIWholesaleAPI#CLASS_DOMAIN
* @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
* @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
15 * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
* @see NSIWholesaleAPI#PARAM_TECH_CONTACT_PASSWORD
* @see NSIWholesaleAPI#PARAM_DOMAIN_NAME
*
* @see WSPResponse#GENERAL_ACCESS_DENIED
20 * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
* @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
*
*/
25 public WSPResponse lookupDomainByTechContact(
    String pCustomerId,
    String pTechContactId,
    String pTechContactPassword,
    String pDomainName)
30 throws NSIWholesaleException {
    return lookupDomainByTechContact(
        pCustomerId,
        pTechContactId,
        pTechContactPassword,
        pDomainName,
        new Properties(),
        0);
35
    }
    /**
40  * Retrieves details of a specific domain product owned by customer,
    * for which the technical contact id is the technical contact for
    * the specific product.
    * <BR><PRE>
        Expected Major Result Code:
45  COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_DOMAIN_NAME
            ATTR_TECH_CONTACT_ID
            ATTR_HOST_NAME1
            ATTR_HOST_ADDR1
50  ATTR_HOST_ID1
            ATTR_HOST_NAME2
            ATTR_HOST_ADDR2
            ATTR_HOST_ID2
55  * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pTechContactId the parameter value

```

```

    * @param pTechContactPassword the parameter value
    * @param pDomainName the parameter value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
5    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DOMAIN
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DOMAIN
10    * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_ID
    * @see NSIWholesaleAPI#PARM_TECH_CONTACT_PASSWORD
    * @see NSIWholesaleAPI#PARM_DOMAIN_NAME
    *
15    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    *
20    */
    public WSPResponse lookupDomainByTechContact (
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
25        String pDomainName,
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return lookupDomainByTechContact (
            pCustomerId,
30            pTechContactId,
            pTechContactPassword,
            pDomainName,
            optionalAttributes,
            0);
35    }
    protected WSPResponse lookupDomainByTechContact (
        String pCustomerId,
        String pTechContactId,
        String pTechContactPassword,
40        String pDomainName,
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
        String methodName = className +
45        ".lookupDomainByTechContact(StringX4, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
50        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
55            Properties allParmsAndAttrs = new Properties();

            // put optional attributes

```



```

        propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
5      pCustomerAccountId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);
        allParmsAndAttrs.put(PARM_TECH_CONTACT_PASSWORD,
      pTechContactPassword);
        allParmsAndAttrs.put(PARM_DOMAIN_NAME, pDomainName);

10      returnCode = conn.makeRequest(COMBO_LOOKUP_DOMAIN,
      allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
15      try {
          pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
20      ResourceException" + re);
        }
        finally {
            return lookupDomainByTechContact(
25              pCustomerAccountId,
              pTechContactId,
              pTechContactPassword,
              pDomainName,
              optionalAttributes,
              count + 1);
30      }
    }
}

35      /**
    * Retrieves a list of all domain names associated with
    * DNS Hosting products owned by customer.
    * <BR><PRE>
        Expected Major Result Code:
40      COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
        DomainName1:
        DomainName2:
        DomainName3:
45      ...etc...
    * </PRE><BR>
    * @param pCustomerAccountId the parameter value
    * @param pCustomerPassword the parameter value
    * @return the Wholesale Server Protocol Server response
50      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
      unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DNS_HOSTING
55      * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
    *

```

```

    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
5      */
    public WSPResponse lookupDnsHosting(
        String pCustomerId,
        String pCustomerPassword)
10    throws NSIWholesaleException {
        return lookupDnsHosting(
            pCustomerId,
            pCustomerPassword,
            new Properties(),
15            0);
    }
    /**
    * Retrieves a list of all domain names associated with
    * DNS Hosting products owned by customer.
20    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
        DomainName1:
25        DomainName2:
        DomainName3:
        ...etc...
    * </PRE><BR>
    * @param pCustomerId the parameter value
30    * @param pCustomerPassword the parameter value
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
35    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DNS_HOSTING
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
40    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
45    */
    public WSPResponse lookupDnsHosting(
        String pCustomerId,
        String pCustomerPassword,
        Properties optionalAttributes)
50    throws NSIWholesaleException {
        return lookupDnsHosting(
            pCustomerId,
            pCustomerPassword,
            optionalAttributes,
55            0);
    }

```

```

        protected WSPResponse lookupDnsHosting(
            String pCustomerId,
            String pCustomerPassword,
            Properties optionalAttributes,
5           int count)
        throws NSIWholesaleException {
            String methodName = className + ".lookupDnsHosting(String, String,
            Properties, int) ";
            if (count > retry) {
10             throw new NSIWholesaleException(methodName + "Unable to
            complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;
15
            try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

                // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);

                allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
20             pCustomerId);
                allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
25             pCustomerPassword);

                returnCode = conn.makeRequest(COMBO_LOOKUP_DNS_HOSTING,
            allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
            } catch (Exception e) {
                try {
30                 pool.release(conn);
                }
                catch (ResourceException re) {
                    System.err.println(methodName + "Caught
            ResourceException" + re);
                }
            finally {
40                 return lookupDnsHosting(
                    pCustomerId,
                    pCustomerPassword,
                    optionalAttributes,
                    count + 1);
            }
        }
    }

50
    /**
     * Retrieves details of a specific
     * DNS Hosting product owned by customer.
     * Used to verify a domain name is part of a DNS Hosting Product
55     * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED

```

```

        Expected Result Property Key(s) Returned:
            ATTR_DOMAIN_NAME
    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
5    * @param pDomainName the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
10    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
    * @see NSIWholesaleAPI#COMBO_LOOKUP_DNS_HOSTING
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
15    * @see NSIWholesaleAPI#PARAM_DOMAIN_NAME
    *
    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see WSPResponse#OBJECT_NOT_FOUND
    * @see
20    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    *
    */
    public WSPResponse lookupDnsHosting(
25        String pCustomerId,
        String pCustomerPassword,
        String pDomainName)
        throws NSIWholesaleException {
        return lookupDnsHosting(
30            pCustomerId,
            pCustomerPassword,
            pDomainName,
            new Properties(),
            0);
35    }
    /**
    * Retrieves details of a specific
    * DNS Hosting product owned by customer.
    * Used to verify a domain name is part of a DNS Hosting Product
40    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_DOMAIN_NAME
45    * </PRE><BR>
    * @param pCustomerId the parameter value
    * @param pCustomerPassword the parameter value
    * @param pDomainName the parameter value
    * @param optionalAttributes a property list of optional attributes
50    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_DNS_HOSTING
55    * @see NSIWholesaleAPI#COMBO_LOOKUP_DNS_HOSTING
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD

```

```

        * @see NSIWholesaleAPI#PARAM_DOMAIN_NAME
        *
        * @see WSPResponse#GENERAL_ACCESS_DENIED
        * @see WSPResponse#OBJECT_NOT_FOUND
5       * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
        *
        */
10     public WSPResponse lookupDnsHosting(
        String pCustomerId,
        String pCustomerPassword,
        String pDomainName,
        Properties optionalAttributes)
15     throws NSIWholesaleException {
        return lookupDnsHosting(
            pCustomerId,
            pCustomerPassword,
            pDomainName,
            optionalAttributes,
20            0);
    }

    protected WSPResponse lookupDnsHosting(
        String pCustomerId,
        String pCustomerPassword,
        String pDomainName,
        Properties optionalAttributes,
        int count)
30     throws NSIWholesaleException {
        String methodName = className + ".lookupDnsHosting(String, String,
String, Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
35        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
40            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);
45            allParmsAndAttrs.put(PARAM_CUSTOMER_ACCOUNT_ID,
pCustomerId);
            allParmsAndAttrs.put(PARAM_CUSTOMER_PASSWORD,
pCustomerPassword);
50            allParmsAndAttrs.put(PARAM_DOMAIN_NAME, pDomainName);

            returnCode = conn.makeRequest(COMBO_LOOKUP_DNS_HOSTING,
allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
55        } catch (Exception e) {
            try {

```

```

        pool.release(conn);
    }
    catch (ResourceException re) {
        System.err.println(methodName + "Caught
5 ResourceException" + re);
    }
    finally {
        return lookupDnsHosting(
            pCustomerAccountId,
10         pCustomerPassword,
            pDomainName,
            optionalAttributes,
            count + 1);
    }
15 }
}

/**
20  * Retrieves a list of all individual accounts linked to a wholesaler.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
25         CustomerAccountId1:
            CustomerAccountId2:
            CustomerAccountId3:
            ...etc...
    * </PRE><BR>
30  * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
35  * @see NSIWholesaleAPI#COMBO_LOOKUP_INDIVIDUAL_ACCOUNT
    *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
40 WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *
    */
    public WSPResponse lookupIndividualAccount()
    throws NSIWholesaleException {
        return lookupIndividualAccount(
45         new Properties(),
            0);
    }
}
/**
50  * Retrieves a list of all individual accounts linked to a wholesaler.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
55         CustomerAccountId1:
            CustomerAccountId2:
            CustomerAccountId3:
            ...etc...

```

```

    * </PRE><BR>
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
5  unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_LOOKUP_INDIVIDUAL_ACCOUNT
    *
10  * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
    WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    */
15  public WSPResponse lookupIndividualAccount(
        Properties optionalAttributes)
        throws NSIWholesaleException {
        return lookupIndividualAccount(
20  optionalAttributes,
        0);
    }
    protected WSPResponse lookupIndividualAccount(
        Properties optionalAttributes,
        int count)
25  throws NSIWholesaleException {
        String methodName = className +
        ".lookupIndividualAccount(Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
30  complete command");
        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

35  try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
40  propertiesCat(allParmsAndAttrs, optionalAttributes);

            returnCode =
            conn.makeRequest(COMBO_LOOKUP_INDIVIDUAL_ACCOUNT, allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
45  } catch (Exception e) {
            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
50  System.err.println(methodName + "Caught
            ResourceException" + re);
            }
            finally {
55  return lookupIndividualAccount(
                optionalAttributes,
                count + 1);
            }
        }
    }

```

```

    }
}

5
/**
 * Retrieves details of a specific individual customer account.
 * <BR><PRE>
    Expected Major Result Code:
10  COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    Expected Result Property Key(s) Returned:
        ATTR_FIRST_NAME
        ATTR_LAST_NAME
        ATTR_PHONE
15        ATTR_AUTH_QUESTION
        ATTR_AUTH_ANSWER
        ATTR_FAX
        ATTR_EMAIL
        ATTR_ADDRESS1
20        ATTR_ADDRESS2
        ATTR_ADDRESS3
        ATTR_ADDRESS4
        ATTR_ADDRESS5
        ATTR_CITY
25        ATTR_STATE
        ATTR_POSTAL_CODE
        ATTR_COUNTRY_CODE
    * </PRE><BR>
    * @param pCustomerId the parameter value
30    * @param pCustomerPassword the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
35    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_LOOKUP_INDIVIDUAL_ACCOUNT
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
    * @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
    *
40    * @see WSPResponse#GENERAL_ACCESS_DENIED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *
45    */
    public WSPResponse lookupIndividualAccount(
        String pCustomerId,
        String pCustomerPassword)
        throws NSIWholesaleException {
        return lookupIndividualAccount(
50            pCustomerId,
            pCustomerPassword,
            new Properties(),
            0);
    }
55 /**
 * Retrieves details of a specific individual customer account.
 * <BR><PRE>

```



```

Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
Expected Result Property Key(s) Returned:
ATTR_FIRST_NAME
ATTR_LAST_NAME
ATTR_PHONE
ATTR_AUTH_QUESTION
ATTR_AUTH_ANSWER
ATTR_FAX
ATTR_EMAIL
ATTR_ADDRESS1
ATTR_ADDRESS2
ATTR_ADDRESS3
ATTR_ADDRESS4
ATTR_ADDRESS5
ATTR_CITY
ATTR_STATE
ATTR_POSTAL_CODE
ATTR_COUNTRY_CODE
* </PRE><BR>
* @param pCustomerId the parameter value
* @param pCustomerPassword the parameter value
* @param optionalAttributes a property list of optional attributes
* @return the Wholesale Server Protocol Server response
* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
* @see NSIWholesaleAPI#COMMAND_LOOKUP
* @see NSIWholesaleAPI#CLASS_INDIVIDUAL_ACCOUNT
* @see NSIWholesaleAPI#COMBO_LOOKUP_INDIVIDUAL_ACCOUNT
* @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
* @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
*
* @see WSPResponse#GENERAL_ACCESS_DENIED
* @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
*/
public WSPResponse lookupIndividualAccount(
String pCustomerId,
String pCustomerPassword,
Properties optionalAttributes)
throws NSIWholesaleException {
return lookupIndividualAccount(
pCustomerId,
pCustomerPassword,
optionalAttributes,
0);
}
protected WSPResponse lookupIndividualAccount(
String pCustomerId,
String pCustomerPassword,
Properties optionalAttributes,
int count)
throws NSIWholesaleException {
String methodName = className + ".lookupIndividualAccount(String,
String, Properties, int) ";
if (count > retry) {

```

```

        throw new NSIWholesaleException(methodName + "Unable to
complete command");
    }
    NSIWholesaleConnection conn = null;
5    WSPResponse returnCode;

    try {
        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();
10        // put optional attributes
        propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
15    pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
        pCustomerPassword);

        returnCode =
20    conn.makeRequest(COMBO_LOOKUP_INDIVIDUAL_ACCOUNT, allParmsAndAttrs);
        pool.checkIn(conn);
        return returnCode;
    } catch (Exception e) {
        try {
25            pool.release(conn);
        }
        catch (ResourceException re) {
            System.err.println(methodName + "Caught
ResourceException" + re);
30        }
        finally {
            return lookupIndividualAccount(
                pCustomerAccountId,
                pCustomerPassword,
35                optionalAttributes,
                count + 1);
        }
    }
}

/**
45    * Retrieves a list of all business customer accounts linked to
    wholesaler.
    * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
50        CustomerAccountId1:
        CustomerAccountId2:
        CustomerAccountId3:
        ...etc...
    * </PRE><BR>
55    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command

```

```

    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_LOOKUP_BUSINESS_ACCOUNT
5      *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *
10     */
    public WSPResponse lookupBusinessAccount()
    throws NSIWholesaleException {
        return lookupBusinessAccount(
            new Properties(),
15            0);
    }
    /**
    * Retrieves a list of all business customer accounts linked to
    wholesaler.
    * <BR><PRE>
20         Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            CustomerAccountId1:
            CustomerAccountId2:
25            CustomerAccountId3:
            ...etc...
    * </PRE><BR>
    * @param optionalAttributes a property list of optional attributes
    * @return the Wholesale Server Protocol Server response
30    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
    unable to complete command
    * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
    * @see NSIWholesaleAPI#COMBO_LOOKUP_BUSINESS_ACCOUNT
35    *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED
    * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
    *
40    */
    public WSPResponse lookupBusinessAccount(
        Properties optionalAttributes)
    throws NSIWholesaleException {
        return lookupBusinessAccount(
45            optionalAttributes,
            0);
    }
    protected WSPResponse lookupBusinessAccount(
        Properties optionalAttributes,
50        int count)
    throws NSIWholesaleException {
        String methodName = className +
        ".lookupBusinessAccount(Properties, int) ";
        if (count > retry) {
55            throw new NSIWholesaleException(methodName + "Unable to
            complete command");
        }
    }

```

```

NSIWholesaleConnection conn = null;
WSPResponse returnCode;

    try {
5         conn = (NSIWholesaleConnection)pool.checkOut();
           Properties allParmsAndAttrs = new Properties();

           // put optional attributes
           propertiesCat(allParmsAndAttrs, optionalAttributes);
10
           returnCode = conn.makeRequest(COMBO_LOOKUP_BUSINESS_ACCOUNT,
allParmsAndAttrs);
           pool.checkIn(conn);
           return returnCode;
15     } catch (Exception e) {
           try {
               pool.release(conn);
           }
           catch (ResourceException re) {
20               System.err.println(methodName + "Caught
ResourceException" + re);
           }
           finally {
25               return lookupBusinessAccount(
                   optionalAttributes,
                   count + 1);
           }
       }
30
    }

    /**
    * Retrieves details of a specific business customer account.
    * <BR><PRE>
35         Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        Expected Result Property Key(s) Returned:
            ATTR_COMPANY_NAME
            ATTR_COMPANY_TYPE
40            ATTR_EMAIL
            ATTR_PHONE
            ATTR_FAX
            ATTR_ADDRESS1
            ATTR_ADDRESS2
            ATTR_ADDRESS3
45            ATTR_ADDRESS4
            ATTR_ADDRESS5
            ATTR_CITY
            ATTR_STATE
50            ATTR_POSTAL_CODE
            ATTR_COUNTRY_CODE
            ATTR_LEGAL_CONTACT_FIRST_NAME
            ATTR_LEGAL_CONTACT_LAST_NAME
            ATTR_LEGAL_CONTACT_PHONE
55            ATTR_LEGAL_CONTACT_FAX
            ATTR_LEGAL_CONTACT_EMAIL
            ATTR_LEGAL_CONTACT_CITY

```

```

ATTR_LEGAL_CONTACT_STATE
ATTR_LEGAL_CONTACT_POSTAL_CODE
ATTR_LEGAL_CONTACT_COUNTRY_CODE
ATTR_LEGAL_CONTACT_ADDRESS1
5      ATTR_LEGAL_CONTACT_ADDRESS2
      ATTR_LEGAL_CONTACT_ADDRESS3
      ATTR_LEGAL_CONTACT_ADDRESS4
      ATTR_LEGAL_CONTACT_ADDRESS5
      * </PRE><BR>
10     * @param pCustomerId the parameter value
      * @param pCustomerPassword the parameter value
      * @return the Wholesale Server Protocol Server response
      * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
15     * @see NSIWholesaleAPI#COMMAND_LOOKUP
      * @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
      * @see NSIWholesaleAPI#COMBO_LOOKUP_BUSINESS_ACCOUNT
      * @see NSIWholesaleAPI#PARM_CUSTOMER_ACCOUNT_ID
      * @see NSIWholesaleAPI#PARM_CUSTOMER_PASSWORD
20     *
      * @see WSPResponse#GENERAL_ACCESS_DENIED
      * @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
      *
25     */
    public WSPResponse lookupBusinessAccount(
        String pCustomerId,
        String pCustomerPassword)
        throws NSIWholesaleException {
30         return lookupBusinessAccount(
            pCustomerId,
            pCustomerPassword,
            new Properties(),
            0);
35     }
    /**
      * Retrieves details of a specific business customer account.
      * <BR><PRE>
          Expected Major Result Code:
40     COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
          Expected Result Property Key(s) Returned:
              ATTR_COMPANY_NAME
              ATTR_COMPANY_TYPE
              ATTR_EMAIL
45             ATTR_PHONE
              ATTR_FAX
              ATTR_ADDRESS1
              ATTR_ADDRESS2
              ATTR_ADDRESS3
50             ATTR_ADDRESS4
              ATTR_ADDRESS5
              ATTR_CITY
              ATTR_STATE
              ATTR_POSTAL_CODE
              ATTR_COUNTRY_CODE
55             ATTR_LEGAL_CONTACT_FIRST_NAME
              ATTR_LEGAL_CONTACT_LAST_NAME

```

```

ATTR_LEGAL_CONTACT_PHONE
ATTR_LEGAL_CONTACT_FAX
ATTR_LEGAL_CONTACT_EMAIL
ATTR_LEGAL_CONTACT_CITY
5   ATTR_LEGAL_CONTACT_STATE
ATTR_LEGAL_CONTACT_POSTAL_CODE
ATTR_LEGAL_CONTACT_COUNTRY_CODE
ATTR_LEGAL_CONTACT_ADDRESS1
ATTR_LEGAL_CONTACT_ADDRESS2
10  ATTR_LEGAL_CONTACT_ADDRESS3
ATTR_LEGAL_CONTACT_ADDRESS4
ATTR_LEGAL_CONTACT_ADDRESS5

* </PRE><BR>
* @param pCustomerId the parameter value
* @param pCustomerPassword the parameter value
15  * @param optionalAttributes a property list of optional attributes
* @return the Wholesale Server Protocol Server response
* @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
20  * @see NSIWholesaleAPI#COMMAND_LOOKUP
* @see NSIWholesaleAPI#CLASS_BUSINESS_ACCOUNT
* @see NSIWholesaleAPI#COMBO_LOOKUP_BUSINESS_ACCOUNT
* @see NSIWholesaleAPI#PARAM_CUSTOMER_ACCOUNT_ID
* @see NSIWholesaleAPI#PARAM_CUSTOMER_PASSWORD
25  *
* @see WSPResponse#GENERAL_ACCESS_DENIED
* @see
WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
30  */
public WSPResponse lookupBusinessAccount(
    String pCustomerId,
    String pCustomerPassword,
    Properties optionalAttributes)
35  throws NSIWholesaleException {
    return lookupBusinessAccount(
        pCustomerId,
        pCustomerPassword,
        optionalAttributes,
40  0);
}
protected WSPResponse lookupBusinessAccount(
    String pCustomerId,
    String pCustomerPassword,
    Properties optionalAttributes,
45  int count)
    throws NSIWholesaleException {
    String methodName = className + ".lookupBusinessAccount(String,
String, Properties, int) ";
50  if (count > retry) {
        throw new NSIWholesaleException(methodName + "Unable to
complete command");
    }
    NSIWholesaleConnection conn = null;
55  WSPResponse returnCode;

    try {

```

```

        conn = (NSIWholesaleConnection)pool.checkOut();
        Properties allParmsAndAttrs = new Properties();

        // put optional attributes
5       propertiesCat(allParmsAndAttrs, optionalAttributes);

        allParmsAndAttrs.put(PARM_CUSTOMER_ACCOUNT_ID,
pCustomerAccountId);
        allParmsAndAttrs.put(PARM_CUSTOMER_PASSWORD,
10      pCustomerPassword);

        returnCode = conn.makeRequest(COMBO_LOOKUP_BUSINESS_ACCOUNT,
allParmsAndAttrs);
        pool.checkIn(conn);
15      return returnCode;
    } catch (Exception e) {
        try {
            pool.release(conn);
        }
        catch (ResourceException re) {
20          System.err.println(methodName + "Caught
ResourceException" + re);
        }
        finally {
25          return lookupBusinessAccount(
                pCustomerAccountId,
                pCustomerPassword,
                optionalAttributes,
                count + 1);
30      }
    }
}

35  /**
    * Retrieves details of a specific technical contact.
    * <BR><PRE>
        Expected Major Result Code:
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
40      Expected Result Property Key(s) Returned:
        ATTR_ORGANIZATION
        ATTR_PHONE
        ATTR_FAX
        ATTR_EMAIL
45      ATTR_ADDRESS1
        ATTR_CITY
        ATTR_STATE
        ATTR_POSTAL_CODE
        ATTR_COUNTRY_CODE
50      * </PRE><BR>
    * @param pTechContactId the parameter value
    * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
55      * @see NSIWholesaleAPI#COMMAND_LOOKUP
    * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
    * @see NSIWholesaleAPI#COMBO_LOOKUP_TECHNICAL_CONTACT

```

```

        * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
        *
        * @see
5      WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        *
        */
        public WSPResponse lookupTechnicalContact(
            String pTechContactId)
            throws NSIWholesaleException {
10          return lookupTechnicalContact(
                pTechContactId,
                new Properties(),
                0);
        }
15      /**
        * Retrieves details of a specific technical contact.
        * <BR><PRE>
            Expected Major Result Code:
            COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
20          Expected Result Property Key(s) Returned:
                ATTR_ORGANIZATION
                ATTR_PHONE
                ATTR_FAX
                ATTR_EMAIL
25          ATTR_ADDRESS1
                ATTR_CITY
                ATTR_STATE
                ATTR_POSTAL_CODE
                ATTR_COUNTRY_CODE
30      * </PRE><BR>
        * @param pTechContactId the parameter value
        * @param optionalAttributes a property list of optional attributes
        * @return the Wholesale Server Protocol Server response
        * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
35      unable to complete command
        * @see NSIWholesaleAPI#COMMAND_LOOKUP
        * @see NSIWholesaleAPI#CLASS_TECHNICAL_CONTACT
        * @see NSIWholesaleAPI#COMBO_LOOKUP_TECHNICAL_CONTACT
        * @see NSIWholesaleAPI#PARAM_TECH_CONTACT_ID
40      *
        * @see
        WSPResponse#COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED
        *
        */
45      public WSPResponse lookupTechnicalContact(
            String pTechContactId,
            Properties optionalAttributes)
            throws NSIWholesaleException {
        return lookupTechnicalContact(
50          pTechContactId,
            optionalAttributes,
            0);
        }
        protected WSPResponse lookupTechnicalContact(
55          String pTechContactId,
            Properties optionalAttributes,
            int count)

```



```

        throws NSIWholesaleException {
            String methodName = className + ".lookupTechnicalContact(String,
Properties, int) ";
            if (count > retry) {
5               throw new NSIWholesaleException(methodName + "Unable to
complete command");
            }
            NSIWholesaleConnection conn = null;
            WSPResponse returnCode;
10
            try {
                conn = (NSIWholesaleConnection)pool.checkOut();
                Properties allParmsAndAttrs = new Properties();

15                // put optional attributes
                propertiesCat(allParmsAndAttrs, optionalAttributes);

                allParmsAndAttrs.put(PARM_TECH_CONTACT_ID, pTechContactId);

20                returnCode =
                conn.makeRequest(COMBO_LOOKUP_TECHNICAL_CONTACT, allParmsAndAttrs);
                pool.checkIn(conn);
                return returnCode;
            } catch (Exception e) {
25                try {
                    pool.release(conn);
                }
                catch (ResourceException re) {
                    System.err.println(methodName + "Caught
30 ResourceException" + re);
                }
                finally {
                    return lookupTechnicalContact(
                        pTechContactId,
                        optionalAttributes,
                        count + 1);
                }
            }
40
        }

/**
 * This method returns basic information about the supported
 * protocol commands and classes.
45 * <BR><PRE>
        Expected Major Result Code:
        COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
        Expected Text Returned:
        ...The Help Text...
50        Expected Result Property Key(s) Returned:
            text1:
            text2:
            text3:
            ...etc...

55 * </PRE><BR>
 * @return the Wholesale Server Protocol Server response

```

```

    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see WSPResponse#getProperties
    * @see NSIWholesaleAPI#COMMAND_HELP
5      *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    *
    */
    public WSPResponse help()
10     throws NSIWholesaleException {
        return help(
            new Properties(),
            0);
    }
15     /**
    * This method returns basic information about the supported
    * protocol commands and classes.
    * <BR><PRE>
        Expected Major Result Code:
20     COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
        Expected Text Returned:
            ...The Help Text...
        Expected Result Property Key(s) Returned:
            text1:
25             text2:
            text3:
            ...etc...
    * </PRE><BR>
    * @param optionalAttributes a property list of optional attributes
30     * @return the Wholesale Server Protocol Server response
    * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
    * @see WSPResponse#getProperties
    * @see NSIWholesaleAPI#COMMAND_HELP
35     *
    * @see WSPResponse#COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    *
    */
    public WSPResponse help(
40         Properties optionalAttributes)
        throws NSIWholesaleException {
        return help(
            optionalAttributes,
            0);
45     }
    protected WSPResponse help(
        Properties optionalAttributes,
        int count)
        throws NSIWholesaleException {
50         String methodName = className + ".help(Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
        }
55         NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

```

```

        try {
            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

5           // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);

            returnCode = conn.makeRequest(COMMAND_HELP,
allParmsAndAttrs);
10           pool.checkIn(conn);
            return returnCode;
        } catch (Exception e) {
            try {
                pool.release(conn);
15           }
            catch (ResourceException re) {
                System.err.println(methodName + "Caught
ResourceException" + re);
20           }
            finally {
                return help(
                    optionalAttributes,
                    count + 1);
            }
25       }
    }

    /**
     * This method terminates the wholesaler's access to session commands.
30     * <BR><PRE>
        Expected Major Result Code: COMMAND_SUCCEEDED_CLOSING_CONNECTION
     * </PRE><BR>
     * @param optionalAttributes a property list of optional attributes
     * @return the Wholesale Server Protocol Server response
35     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
     * @see NSIWholesaleAPI#COMMAND_QUIT
     *
     * @see WSPResponse#COMMAND_SUCCEEDED_CLOSING_CONNECTION
40     */
    public WSPResponse quit()
    throws NSIWholesaleException {
        return quit(
45           new Properties(),
            0);
    }

    /**
     * This method terminates the wholesaler's access to session commands.
50     * <BR><PRE>
        Expected Major Result Code: COMMAND_SUCCEEDED_CLOSING_CONNECTION
     * </PRE><BR>
     * @param optionalAttributes a property list of optional attributes
     * @return the Wholesale Server Protocol Server response
55     * @exception nsi.wholesale.toolkit.javaapi.NSIWholesaleException when
unable to complete command
     * @see NSIWholesaleAPI#COMMAND_QUIT

```

```

    *
    * @see WSPResponse#COMMAND_SUCCEEDED_CLOSING_CONNECTION
    *
    */
5   public WSPResponse quit(
        Properties optionalAttributes)
    throws NSIWholesaleException {
        return quit(
10         optionalAttributes,
            0);
    }
    protected WSPResponse quit(
        Properties optionalAttributes,
        int count)
15    throws NSIWholesaleException {
        String methodName = className + ".quit(Properties, int) ";
        if (count > retry) {
            throw new NSIWholesaleException(methodName + "Unable to
complete command");
20        }
        NSIWholesaleConnection conn = null;
        WSPResponse returnCode;

        try {
25            conn = (NSIWholesaleConnection)pool.checkOut();
            Properties allParmsAndAttrs = new Properties();

            // put optional attributes
            propertiesCat(allParmsAndAttrs, optionalAttributes);
30            returnCode = conn.makeRequest(COMMAND_QUIT,
allParmsAndAttrs);
            pool.checkIn(conn);
            return returnCode;
35        } catch (Exception e) {
            try {
                pool.release(conn);
            }
            catch (ResourceException re) {
40                System.err.println(methodName + "Caught
ResourceException" + re);
            }
            finally {
                return quit(
45                    optionalAttributes,
                        count + 1);
            }
        }
    }
}
50 //##--END API-----
-----

/**
 * This method shutdowns the Wholesaler Services Protocol API.
55  * @return void
 *
 */

```

```

    public void shutdown() {
        System.out.println("NSIWholesaleAPI.shutdown: " +
            "Calling pool.shutdown()...");
        //pool.shutdown();
5       try {
            //pool.close();
            pool.close(60000, 60000);
        }
        catch (ResourceException re) {
10         System.err.println("Exception Caught: " + re);
        }
        System.out.println("NSIWholesaleAPI.shutdown: " +
            "pool.shutdown() returned!");
15    }

//*****
// Support Functions

20    /**
     * This method parses the string argument as a signed decimal integer
     * @param valueString the string to be parsed as an int
     * @return an int representing the valueString
     * @see java.lang.Integer#parseInt
     *
25    */
    protected static int parseInt(String valueString) {
        try {
            return Integer.parseInt(valueString);
        } catch (NumberFormatException nfe) {
30            return 0;
        }
    }

    /**
35     * This method concatenates the source property list onto the end of the
     * target property list. Filtering is done to remove properties
    associated with
     * credit card charges.
     * @param target the target (destination) property list
40     * @param source the source property list
     * @return void
     * @see java.util.Properties
     *
45    */
    protected static void propertiesCat(Properties target, Properties
    source) {
        String methodName = className + ".propertiesCat(Properties,
        Properties) ";

50        if ((target == null) || (source == null)) {
            return;
        }

        // concatenate source properties onto target properties
55        if (source.isEmpty() == false) {
            try {
                String key;

```

```

        String value;
        Enumeration e = source.propertyNames();
        while (e.hasMoreElements()) {
            key = (String)e.nextElement();
            value = source.getProperty(key);
            target.put(key, value);
        }
    }
    catch (NoSuchElementException nsee) {
        System.err.println(methodName +
            "Invalid Properties");
        nsee.printStackTrace();
    }
}
return;
}

/**
 * This method concatenates the source property list onto the end of the
 * target property list.
 * @param protocol client (your) protocol
 * @return true if your protocol matches wholesale services protocol,
 * otherwise false
 * @see NSIWholesaleAPI#ATTR_PROTOCOL
 * @see NSIWholesaleAPI#Describe
 */
protected boolean verifyConnectionProtocol(String protocol)
throws NSIWholesaleException {
    String methodName = className + ".verifyConnectionProtocol(String)";

    WSPResponse wspResponse = describe(ATTR_PROTOCOL);
    if (wspResponse == null) {
        System.err.println(methodName + "Unable to verify connection
protocol " + protocol);
        return false;
    }
    if (wspResponse.getMajorResultCode() ==
WSPResponse.COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED) {
        String wspProtocol = wspResponse.getProperty(ATTR_PROTOCOL);
        if (wspProtocol == null) {
            System.err.println(methodName + "Unable to verify
connection protocol " + protocol);
            return false;
        }
        if (protocol.equals(wspProtocol)) {
            System.out.println(methodName + "Successfully verified
connection protocol " + protocol);
            return true;
        }
        else {
            System.err.println(methodName + "Failed to verify
connection protocol: " +
                "Your Protocol = <" + protocol + "> while
Wholesale Services Protocol = <" + wspProtocol + ">");
            return false;
        }
    }
}

```

```

        }
        else {
            System.err.println(methodName + "Unable to verify connection
5 protocol " + protocol);
            return false;
        }
    }
}
10 //-----
// End of file NSIWholesaleAPI.java
// ----- Copyright (c) 2000 Network Solutions, Inc -----

```

The class definition for WSPResponse is as follows:

```

//-----
// Package
20 //-----
package nsi.wholesale.toolkit.javaapi;

//-----
// Imports...
25 //-----
import java.util.NoSuchElementException;
import java.util.StringTokenizer;
import java.util.Hashtable;
import java.util.Vector;
30 import java.util.Enumeration;
import java.util.Properties;
import java.text.*;

/**
35 * ----- Copyright (c) 2000 Network Solutions, Inc -----<BR>
* This class represents the various components of a
* Wholesale Services Protocol (WSP) Response.
* A WSP response will be returned for each request that is issued
* through the protocol. A response will describe, in as much detail
40 * as possible, the outcome of the issued request, and will optionally
* include any additional information that was generated by the request.
* A result consists of a 'result code' followed by a descriptive
* 'result message'. A 'result code' consists of a major result code
* followed by a period ('.'), followed by a minor result code.
45 * The three-digit major result code provides a general indication of
* the success or failure of an issued request.
* In most cases, one or more result properties are returned. A result
* property consists of a property key followed by a colon (':'),
* followed by a property value.
50 * In other cases, plain text is returned (e.g. the information returned
* from the Help command). Both of these cases are indicated by specific
* result codes.
*
* @author      David A. Moston
55 * @see        NSIWholesaleAPI

```

```

* @since      27 July 2000
* @version    1.0
*
*
5  *<BR>-----<BR>
* $Log$
*<BR>-----<BR>
*/
10 public class WSPResponse {
//-----
// Beginning of Required/Standard Fields
//-----
15     /** Serial Version */
        static long serialVersion = 1L;

        /** Release Version */
        static double version = 1.0D;

20     /** SCCS control ID */
        static final String sccsID = "@(#) wholesale-sdk_d1.12.1.3 08/04/00
10:27:15 WSPResponse.java NSI";

        /** JDK Version */
25     static String jdkVersion = "1.1.7";

        // Static Accessor Method(s)...
        /** Static accessor method to get serial version */
        static public long getSerialVersion() { return serialVersion; }
30
        /** Static accessor method to get release version */
        static public double getVersion() { return version; }

        /** Static accessor method to get JDK version */
35     static public String getJdkVersion() { return jdkVersion; }
//-----
// End of Required/Standard Fields
//-----
40
//-----
// class (static) variable(s) ...
//-----
        /** The name of this class */
        private static final String className = "WSPResponse";
45

        /** Ease-of-use constant for Major Result Code: Minimum.
        (Note: this is NOT a valid/supported major result code)
        All valid/supported major result codes will be
        greater than the minimum and less than the maximum, respectively */
50     public static final int MAJOR_RESULT_CODE_MIN = 99;

        /** Ease-of-use constant for Major Result Code[General Information]:
        Minimum.
        (Note: this is NOT a valid/supported major result code) */
55     public static final int MAJOR_RESULT_CODE_GENERAL_INFORMATION_MIN = 99;
        /** Ease-of-use constant for Major Result Code[General Information]:
        General information */

```



```

    public static final int GENERAL_INFORMATION = 100;
    /** Ease-of-use constant for Major Result Code[General Information]:
Maximum.
    (Note: this is NOT a valid/supported major result code) */
5     public static final int MAJOR_RESULT_CODE_GENERAL_INFORMATION_MAX = 199;

    /** Ease-of-use constant for Major Result Code[Success]: Minimum.
    (Note: this is NOT a valid/supported major result code) */
    public static final int MAJOR_RESULT_CODE_SUCCESS_MIN = 199;
10    /** Ease-of-use constant for Major Result Code[Success]: Authentication
successful */
    public static final int AUTHENTICATION_SUCCESSFUL = 210;
    /** Ease-of-use constant for Major Result Code[Success]: Session created
*/
15    public static final int SESSION_CREATED = 220;
    /** Ease-of-use constant for Major Result Code[Success]: Command
succeeded, closing connection */
    public static final int COMMAND_SUCCEEDED_CLOSING_CONNECTION = 221;
    /** Ease-of-use constant for Major Result Code[Success]: Command
20    accepted, now executing */
    public static final int COMMAND_ACCEPTED_NOW_EXECUTING = 230;
    /** Ease-of-use constant for Major Result Code[Success]: Command not
available */
    public static final int COMMAND_NOT_AVAILABLE = 231;
25    /** Ease-of-use constant for Major Result Code[Success]: Command
execution succeeded, no data being returned; Ok */
    public static final int
COMMAND_EXECUTION_SUCCEEDED_NO_DATA_BEING_RETURNED = 240;
    /** Ease-of-use constant for Major Result Code[Success]: Command
30    execution succeeded, properties being returned */
    public static final int
COMMAND_EXECUTION_SUCCEEDED_PROPERTIES_BEING_RETURNED = 241;
    /** Ease-of-use constant for Major Result Code[Success]: Command
execution succeeded, text being returned */
35    public static final int COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
= 242;
    /** Ease-of-use constant for Major Result Code[Success]: Maximum.
    (Note: this is NOT a valid/supported major result code) */
    public static final int MAJOR_RESULT_CODE_SUCCESS_MAX = 299;
40

    /** Ease-of-use constant for Major Result Code[Failure]: Minimum.
    (Note: this is NOT a valid/supported major result code) */
    public static final int MAJOR_RESULT_CODE_FAILURE_MIN = 299;
    /** Ease-of-use constant for Major Result Code[Failure]: Authentication
45    failed */
    public static final int AUTHENTICATION_FAILED = 310;
    /** Ease-of-use constant for Major Result Code[Failure]: Too many
connections */
    public static final int TOO_MANY_CONNECTIONS = 320;
50    /** Ease-of-use constant for Major Result Code[Failure]: Too many
connections from current wholesaler */
    public static final int TOO_MANY_CONNECTIONS_FROM_CURRENT_WHOLESALE =
321;
    /** Ease-of-use constant for Major Result Code[Failure]: Wholesaler
55    account has been suspended */
    public static final int WHOLESALE_ACCOUNT_HAS_BEEN_SUSPENDED = 322;

```

```

    /** Ease-of-use constant for Major Result Code[Failure]: Session timed
out */
    public static final int SESSION_TIMED_OUT = 323;
    /** Ease-of-use constant for Major Result Code[Failure]: Server closing
5 connection - Command timed out */
    public static final int SERVER_CLOSING_CONNECTION = 324;
    /** Ease-of-use constant for Major Result Code[Failure]: Invalid command
*/
    public static final int INVALID_COMMAND = 330;
10    /** Ease-of-use constant for Major Result Code[Failure]: Invalid class
*/
    public static final int INVALID_CLASS = 331;
    /** Ease-of-use constant for Major Result Code[Failure]: Formatting
error */
15    public static final int FORMATTING_ERROR = 332;
    /** Ease-of-use constant for Major Result Code[Failure]: Missing
required parameter */
    public static final int MISSING_REQUIRED_PARAMETER = 333;
    /** Ease-of-use constant for Major Result Code[Failure]: Invalid
20 attribute */
    public static final int INVALID_ATTRIBUTE = 334;
    /** Ease-of-use constant for Major Result Code[Failure]: Invalid value
*/
    public static final int INVALID_VALUE = 335;
25    /** Ease-of-use constant for Major Result Code[Failure]: Missing
required attribute */
    public static final int MISSING_REQUIRED_ATTRIBUTE = 336;
    /** Ease-of-use constant for Major Result Code[Failure]: Missing
required value */
30    public static final int MISSING_REQUIRED_VALUE = 337;
    /** Ease-of-use constant for Major Result Code[Failure]: Missing
parameter */
    public static final int INVALID_PARAMETER = 338;
    /** Ease-of-use constant for Major Result Code[Failure]: Command failed,
35 unknown error - Command failed, internal error */
    public static final int COMMAND_FAILED_UNKNOWN_ERROR = 340;
    /** Ease-of-use constant for Major Result Code[Failure]: Object not
found */
    public static final int OBJECT_NOT_FOUND = 341;
40    /** Ease-of-use constant for Major Result Code[Failure]: Command access
denied */
    public static final int COMMAND_ACCESS_DENIED = 342;
    /** Ease-of-use constant for Major Result Code[Failure]: General access
denied - Access to request denied */
45    public static final int GENERAL_ACCESS_DENIED = 343;
    /** Ease-of-use constant for Major Result Code[Failure]: Maximum.
(Note: this is NOT a valid/supported major result code) */
    public static final int MAJOR_RESULT_CODE_FAILURE_MAX = 399;

50    /** Ease-of-use constant for Major Result Code: Maximum.
(Note: this is NOT a valid/supported major result code)
All valid/supported major result codes will be
greater than the minimum and less than the maximum, respectively */
55    public static final int MAJOR_RESULT_CODE_MAX = 400;

//-----

```

```

// class (static) method(s) ...
//-----

//-----
5 // static initializer block ...
//-----

//-----
10 // instance Variable(s) ...
//-----
    /**
     * WSP response result code (consisting of a major result code,
     * followed by a period ('.'), followed by a minor result code).
     */
15 private String resultCode = null;

    /**
     * WSP response major result code
     */
20 private int majorResultCode = 0;

    /**
     * WSP response minor result code
     */
25 private int minorResultCode = 0;

    /**
     * The original WSP text response string used to instantiate object.
     * The raw WSP response consists of one or more <CRLF> delimited lines,
     * terminated by a line containing only a period ('.'), followed by
30 <CRLF>.
     */
    private String rawResponse = null;

35 /**
     * The result message extracted from the original WSP text response.
     */
    private String resultMessage = null;

40 /**
     * The WSP response attributes extracted from the original WSP
     * text response.
     */
45 private Hashtable attributes = null;

//-----
// Constructors
//-----

50 /**
     *
     * Parses a WSP text response and extracts and stores the
     * various components.
55 *
     * @param data the raw WSP response text
     *

```

```

5      * @exception IllegalArgumentException if <code>data</code> is null.
      * @exception InvalidResponseFormatException if <code>data</code> is
      *                                           not a properly formatted
      *                                           WSP response.
      */
      public WSPResponse (String data)
          throws IllegalArgumentException, InvalidResponseFormatException {

10          if (data == null) {
              throw new IllegalArgumentException("null argument: data");
          }

          parseRawResponse(data);
          setRawResponse(data);

15      }
      /**
      * Create a new WSPResponse based on value inputs.
      *
20      * @param majorCode the major code for the WSPResponse.
      * @param minorCode the minor code for the WSPResponse.
      * @param message the text string associated with the message.
      * @param assocProps the properties associated with this message.
      *
25      * @exception IllegalArgumentException if majorCode or minorCode is or
      *                                           not in the appropriate range.
      */
      public WSPResponse (int majorCode, int minorCode,
30                          String message, Properties assocProps)
          throws IllegalArgumentException {

          if ((majorCode == null) || (minorCode == null) ||
              (majorCode < MAJOR_RESULT_CODE_MIN) || (majorCode >
35      MAJOR_RESULT_CODE_MAX) ||
              (minorCode < 0) || (minorCode > 999)) {
              throw new IllegalArgumentException("null argument: data");
          }
          majorResultCode = majorCode;
          minorResultCode = minorCode;
          resultMessage = new String(message);
          if (assocProps != null) {
              Enumeration enum = assocProps.propertyNames();
              while (enum.hasMoreElements()) {
45                  String prop_key = (String) enum.nextElement();
                  String value = prop.getProperty(prop_key);
                  addAttribute(prop_key, value);
              }
          }

50      DecimalFormat myFmt = new DecimalFormat();
          myFmt.setMinimumIntegerDigits(3);
          myFmt.setMaximumIntegerDigits(3);
          myFmt.setMaximumFractionDigits(3);
          myFmt.setMinimumFractionDigits(3);
          double x = majorCode + minorCode/1000.0;
          resultCode = myFmt.format(x);
55

```

```

    }

//-----
// Accessors and Mutators
//-----

    /**
     * Returns the original WSP text response string used to instantiate
     * object.
    10    * @return the original WSP text response.
     */
    public String getRawResponse () {
    15        return rawResponse;
    }

    /**
     * Returns the result message extracted from the original WSP
     * text response.
    20    * @return the WSP response result message.
     */
    public String getResultMessage () {
    25        return resultMessage;
    }

    /**
     * Returns any attributes, if applicable, extracted from the original
    30    * WSP text response.
     * @return response attributes if any exist, null otherwise.
     */
    Hashtable getAttributes () {
    35        return attributes;
    }

    /**
     * returns the result property value corresponding to the specified
    40    * result property key, if found, extracted from the original
     * WSP text response.
     * @param resultPropertyKey a result property key
     * @return the corresponding result property value, otherwise null
    45    */
    public String getProperty (String resultPropertyKey) {
        String resultPropertyValue = null;
        Hashtable attributes = getAttributes();

    50        if (attributes == null) {
            return resultPropertyValue;
        }

        Vector vector = (Vector) attributes.get(resultPropertyKey);

    55        if (vector == null) {
            return resultPropertyValue;
        }
    }

```

```

    }

    for (int i = 0; i < vector.size(); i++) {
        resultPropertyValue = (String)vector.elementAt(i);
5      }

    return resultPropertyValue;
}

10  /**
    * Returns a property list containing all result property
    specifications,
    * if applicable, extracted from the original WSP text response.
    * Note: when the major result code =
15  COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    * text is being returned in the response, as opposed to result property
    specifications.
    * Since there are no defined result property keys for returned text, a
    dummy
20  * result property key is generated for each line of text.
    * That dummy result property key will look like 'text#', where #
    * is a number from 1 to N (N being the number of lines of text
    returned).
    * For example, the first dummy result property key will be 'text1'.
25  * For example, the tenth dummy result property key will be 'text10'.
    *
    *
    * @return response attributes if any exist, null otherwise.
    * @see WSPResponse#getText
30  */
    public Properties getProperties () {
        Properties props = new Properties();

        Hashtable attributes = getAttributes();

35        if (attributes == null) {
            return null;
        }

40        Enumeration enum = attributes.keys();

        int lineNumber = 0;
        while (enum.hasMoreElements()) {
            lineNumber++;
            String key = null;
            Vector vector = null;

45            key = (String) enum.nextElement();
            //if Command execution succeeded; text being returned
            if (getMajorResultCode() ==
50  COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED) {
                key = "text" + lineNumber;    // Create a "dummy" key
                for this line of text
            }
            vector = (Vector) attributes.get(key);

55            for (int i=0; i < vector.size(); i++) {

```

```

        props.put(key, (String)vector.elementAt(i));
    }
}

5      return props;
    }

    /**
    * Returns a property list containing all result property
10 specifications,
    * if applicable, extracted from the original WSP text response..
    * Note: when the major result code =
    COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED
    * text is being returned in the response, as opposed to result property
15 specifications.
    * Since there are no defined result property keys for returned text, a
    dummy
    * result property key is generated for each line of text.
    * That dummy result property key will look like 'text#', where #
20 * is a number from 1 to N (N being the number of lines of text
    returned).
    * For example, the first dummy result property key will be 'text1'.
    * For example, the tenth dummy result property key will be 'text10'.
    *
25 * @param wspResponse the WSP response
    * @return response attributes if any exist, null otherwise.
    * @see WSPResponse#getText
    */
    public static Properties getProperties (WSPResponse wspResponse) {
30      return wspResponse.getProperties();
    }

    /**
    * returns a vector containing lines of text returned in the WSP
35 response.
    * Note: when the major result code is
    COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED,
    * text is being returned in the response, as opposed to result property
    specifications.
40 * There are no result property keys defined for text.
    * This method returns a vector of Strings. Each element (String) in
    the vector is a line of text.
    *
    * @param wspResponse the WSP response
45 *
    * @return a vector containing lines of text returned, otherwise null
    * @see WSPResponse#getProperties
    */
    public Vector getText () {
50      Vector text = new Vector();

      //if not: Command execution succeeded; text being returned
      if (getMajorResultCode() !=
    COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED) {
55      return null;
      }
    }

```

```

        Hashtable attributes = getAttributes();

        if (attributes == null) {
5           return null;
        }

        Enumeration enum = attributes.keys();

        int lineNumber = 0;
10       while (enum.hasMoreElements()) {
            lineNumber++;
            String key = null;
            Vector vector = null;

15             key = (String) enum.nextElement();
            //Command execution succeeded; text being returned
            key = "text" + lineNumber;    // Create a "dummy" key for
this line of text
            vector = (Vector) attributes.get(key);
20             for (int i=0; i < vector.size(); i++) {
                text.addElement((String)vector.elementAt(i));
            }
        }
25       return text;
    }

30     /**
     * sets the raw response text
     *
     * @param aRawResponse the raw WSP response text
     * @return void
35     */
    private void setRawResponse (String aRawResponse) {
        rawResponse = aRawResponse;
    }

40     /**
     * sets the response result code
     *
     * @param inResultCode the WSP response result code
     * @return void
45     */
    private void setResultCode (String inResultCode) {
        resultCode = inResultCode;
    }

50     /**
     * returns the WSP response result code and result message
     *
     * @return the WSP response result code and result message
55     */
    public String getResultCodeAndMessage () {
        return (getResultCode() + " " + getResultMessage());
    }

```



```

    }

    /**
    * returns the WSP response result code
5     *
    * @return the WSP response result code
    */
    public String getResultCode () {
10         return resultCode;
    }

    /**
    * returns the WSP response major result code
    *
15     * @return the WSP response major result code
    */
    public int getMajorResultCode () {
        parseResultCode();
        return majorResultCode;
20    }

    /**
    * returns the WSP response minor result code
    *
25     * @return the WSP response minor result code
    */
    public int getMinorResultCode () {
        parseResultCode();
        return minorResultCode;
30    }

    /**
    * sets the WSP response result message
35     *
    * @param aResultMessage the WSP response result message
    * @return void
    */
    private void setResultMessage (String aResultMessage) {
40         resultMessage = aResultMessage;
    }

    /**
    * sets the WSP response attributes
45     *
    * @param hash the WSP response attributes
    * @return void
    */
50    private void setAttributes (Hashtable hash) {
        attributes = hash;
    }

55    //-----
    // General Methods
    //-----

```

```

5  /**
   * parses the current result code into major and minor result codes.
   *
   * @return true if result code is successfully parsed, otherwise false
   */
   private boolean parseResultCode () {
       boolean rc = true;
       String methodName = className + ".parseResultCode() ";
10      String sResultCode = getResultCode();
       String sMajorResultCode = null;
       String sMinorResultCode = null;
       int iMajorResultCode = 0;
       int iMinorResultCode = 0;

15      try {
           StringTokenizer st = new StringTokenizer(sResultCode, ".");

           sMajorResultCode = st.nextToken();
20          sMinorResultCode = st.nextToken();

           iMajorResultCode = Integer.parseInt(sMajorResultCode);
           iMinorResultCode = Integer.parseInt(sMinorResultCode);
       }
25      catch(NoSuchElementException nsee) {
           System.err.println(methodName + "Unable to parse result code
   <"
           + sResultCode + ">");
           nsee.printStackTrace();
30      }
       catch(NumberFormatException nfe) {
           System.err.println(methodName + "Invalid result code <"
           + sResultCode + ">");
           nfe.printStackTrace();
35      }
       finally {
           if (iMajorResultCode <= 0) {
               rc = false;
           }
40          else {
               majorResultCode = iMajorResultCode;
               minorResultCode = iMinorResultCode;
           }
45      }

       return rc;
   }

   /**
50  * prints debug-type representation of the WSP Response to standard
   output
   *
   * @param response a WSP response
   * @return void
55  */
   public static void displayResponse (WSPResponse response) {
       System.out.println("Result Code: " + response.getResultCode());

```

```

        System.out.println("Major Result Code: " +
            response.getMajorResultCode());
        System.out.println("Minor Result Code: " +
            response.getMinorResultCode());
5       System.out.println("Result Message: " +
response.getResultMessage());

        Hashtable attributes = response.getAttributes();

10      if (attributes == null) {
            System.out.println("Result Properties: NONE");
            return;
        }

15      System.out.println("Result Properties:");

        Enumeration enum = attributes.keys();

        while (enum.hasMoreElements()) {
20            String key = null;
            Vector vector = null;

            key = (String) enum.nextElement();
            vector = (Vector) attributes.get(key);

25            for (int i=0; i < vector.size(); i++) {
                System.out.println(key + ":" +
(String)vector.elementAt(i));
            }
30        }

    }

    /**
35     * returns the raw WSP response
     *
     * @return a string representing the raw WSP response
     */
    public String toString () {
40        if (rawResponse != null) {
            return rawResponse;
        }
        if (responseCode == null) {
            fabricatedResponse = new String("001.001");
45        } else {
            fabricatedResponse = new String(responseCode);
        }
        if (resultMessage == null) {
            fabricatedResponse = new
50 String(fabricatedResponse+"<<Message is null>>\r\n");
        } else {
            fabricatedResponse = new
String(fabricatedResponse+resultMessage+"\r\n");
        }
55        if (attributes != null) {
            Enumeration enum = attributes.keys();

```

```

        while (enum.hasMoreElements()) {
            String key = null;
            Vector vector = null;

5            key = (String) enum.nextElement();
            vector = (Vector) attributes.get(key);

            for (int i=0; i < vector.size(); i++) {
                fabricatedResponse = new
10 String(fabricatedResponse+ key +
                                           ":" +
                                           (String)vector.elementAt(i) +
                                           "\r\n");
15             }
        }
        fabricatedResponse = new String(fabricatedResponse+"\r\n");
        return fabricatedResponse;
20    }

    /**
    * Parse raw WSP text response and extract and store all components
25    *
    * @param data the raw WSP text response
    * @return void
    * @exception InvalidResponseFormatException upon detection of an
    invalid response format
30    */
    private void parseRawResponse (String data)
        throws InvalidResponseFormatException {
        String methodName = className + ".parseRawResponse(String) ";

35        //
        // Split data by <CRLF>
        //
        StringTokenizer st = new StringTokenizer(data, "\r\n");

40        //
        // Get the first line of the response
        //
        String responseLine = "";

45        try {
            responseLine = st.nextToken();

        } catch (NoSuchElementException e) {
50            throw new InvalidResponseFormatException("Missing first
line");
        }

55        //
        // Extract "result code" and "result message" from first line
        // of response.

```

```

//
int spaceIndex = responseLine.indexOf(' ');

5 //
// If there is no space in the string, then the format is invalid.
// First line must be "code message"
//
10 if (spaceIndex == -1) {
    throw new InvalidResponseFormatException(
        "missing/invalid code and message");
    }

//
15 // The code is comprised of the set of characters spanning the
// the first character to the space
//
String myresultCode = responseLine.substring(0, spaceIndex);

20 //
// trim any whitespace from both ends of code
//
myresultCode = myresultCode.trim();

25 //
// Attempt to convert the string result code to an integer
//
setResultCode(myresultCode);

30 //
// Extract the result message now
//
String desc = null;

35 //
// The result message is comprised of all characters spanning from
// the first character after the space to the end of the entire
string
40 desc = responseLine.substring(spaceIndex + 1,
responseLine.length());

//
45 // trim any whitespace from both ends of code
//
desc = desc.trim();

if (desc.length() == 0) {
50     throw new InvalidResponseFormatException(
        "Missing result message");
    }

setResultMessage(desc);

55

String key;

```

```

String value;
int colonIndex = 0;

//if Command execution succeeded; text being returned
5      if (getMajorResultCode() ==
COMMAND_EXECUTION_SUCCEEDED_TEXT_BEING_RETURNED) {
        int lineNumber = 0;
        try {
10            while (st.hasMoreTokens() ) {
                lineNumber++;

                //
                // Create a "dummy" key for this line of text
                //
15            key = "text" + lineNumber;

                //
                // Get the next line from the response
                //
20            value = st.nextToken();

                //
                // Break from the loop if we hit the response
                //
25            if (value.trim().equals(".")) {
                    break;
                }

                //
                // Save attribute
                //
                //addAttribute(key.toLowerCase(), value);
                addAttribute(key, value);
35            } // end while

        } // end try
        catch (NoSuchElementException e) {
40            throw new InvalidResponseFormatException(
                "Error splitting response by '\r\n'", e);
        } // end catch

    }
    else
    {
        //
        // Iterate through the rest of the response, searching for
        // attributes. Break from the loop when we encounter the
        // response terminator
        //
        try {
55            while (st.hasMoreTokens() ) {
                //
                // Get the next line from the response
                //

```

```

        responseLine = st.nextToken();

        //
        // Try to find the colon (':') in the string
5        //
        colonIndex = responseLine.indexOf(':');

        //
        // If the colon is missing, then check to see if
10        we've
        haven't,
        // encountered the response terminator. If we
        // then the response is formatted improperly
        //
15        if (colonIndex == -1) {
            responseLine = responseLine.trim();

            //
            // Break from the loop if we hit the
20        response terminator
            //
            if (responseLine.equals(".")) {
                break;
            }
            else {
25                throw new
                InvalidResponseFormatException(
                    "Invalid attribute key-value
                    pair");
30            }
        }

        //
        // The attribute key is everything before the
35        colon
        //
        key = responseLine.substring(0, colonIndex);
        key = key.trim();

        //
        // The attribute value is everything after the
40        colon
        //
        value = responseLine.substring(colonIndex + 1);
45        value = value.trim();

        //
        // Save attribute
        //
50        //addAttribute(key.toLowerCase(), value);
        addAttribute(key, value);

    }    // end while

55    }    // end try
    catch (NoSuchElementException e) {
        throw new InvalidResponseFormatException(

```

```

        "Error splitting response by '\r\n'", e);
    }    // end catch

    }

5    }//parseRawResponse

    /**
10    * Attributes are stored in a hashtable. Since there can be multiple
    * values for the same key, we store all values in vectors. Everytime
    * a value is added for a key that already exists, we just add that
    * value to that key's vector. If a key does not yet exist, then
    * we create a new vector and add the value to it.
15    *
    * @param key a result property key
    * @param value a result property value
    * @return void
    */
20    private void addAttribute (String key, String value) {
        Hashtable hash = getAttributes();

        if (hash == null) {
            hash = new Hashtable();
25            setAttributes(hash);
        }

        Vector vector = (Vector) hash.get(key);

30        if (vector == null) {
            vector = new Vector(1);
            hash.put(key, vector);
        }

35        vector.addElement(value);
    }

40    }

    //-----
    // End of file WSPResponse.java
45    // ----- Copyright (c) 2000 Network Solutions, Inc -----

```

50

WHAT IS CLAIMED IS:

1. A method for interfacing a user application program to a system for conducting a transaction associated with managing Internet domain names, comprising the steps of:

5 receiving a request in an application program interface from the
 application program to establish over the Internet a session
 between the user and the system;
 authenticating the right of the user to conduct a transaction associated with
 managing Internet domain names;
10 establishing a secure socket layer within the application program interface
 and the system;
 receiving a request in the application program interface from the
 applications program to initiate a transaction following such
 authentication; and
15 transmitting commands from the application program interface to the
 system effectuating the transaction.

2. A method as set forth in claim 1, further comprising the step of:
 notifying the application program as to the status of processing of the
 command.

3. A method as set forth in claim 2, wherein:
the notifying step comprises the step of returning a response to the application program containing a statement indicating whether the transaction has been fulfilled.
4. A method as set forth in claim 3, wherein:
the statement comprises a message which includes an ASCII text portion, followed by a delimiter portion.
5. A method as set forth in claim 4, wherein:
the ASCII text portion is selected from a predefined dictionary of statements.
6. A method as set forth in claim 1, wherein:
authenticating the user is determined from the Internet Protocol address, secure socket layer encryption, and user name and password provided by the application program.
7. A method as set forth in claim 1, wherein:
the authenticating step includes the step of at least one additional attempt to authenticate the user following a first failed attempt.

8. A method as set forth in claim 7, wherein:

following said at least one additional attempt to authenticate the user is denied
further attempts to be authenticated should said at least one additional attempt fail.

9. Apparatus for interfacing a user application program to a system for conducting
transactions associated with managing Internet Domain Names comprising:

a component for receiving a request from an applications program to
establish over the Internet a connection between a user and the
system;

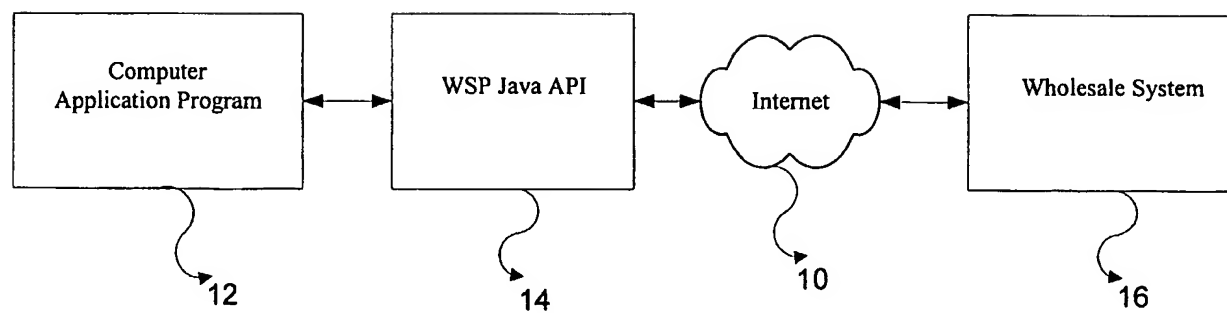
a component for authenticating the right of the user to conduct a
transaction associated with managing Internet domain names;

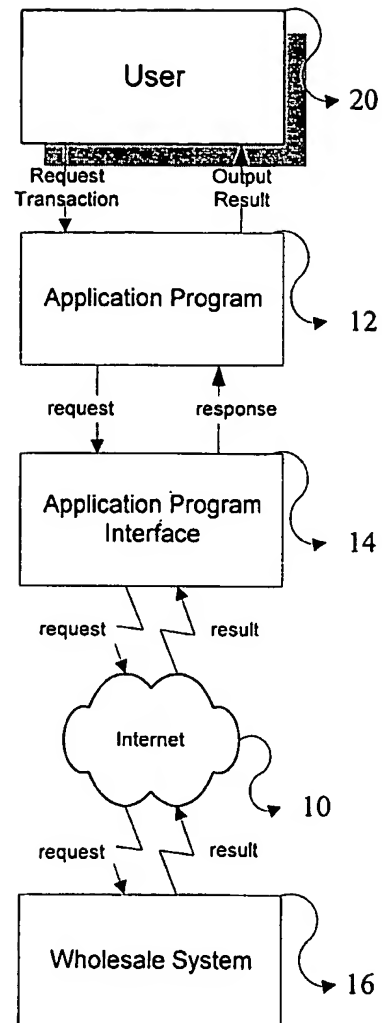
a component for establishing a secure socket layer within the apparatus;

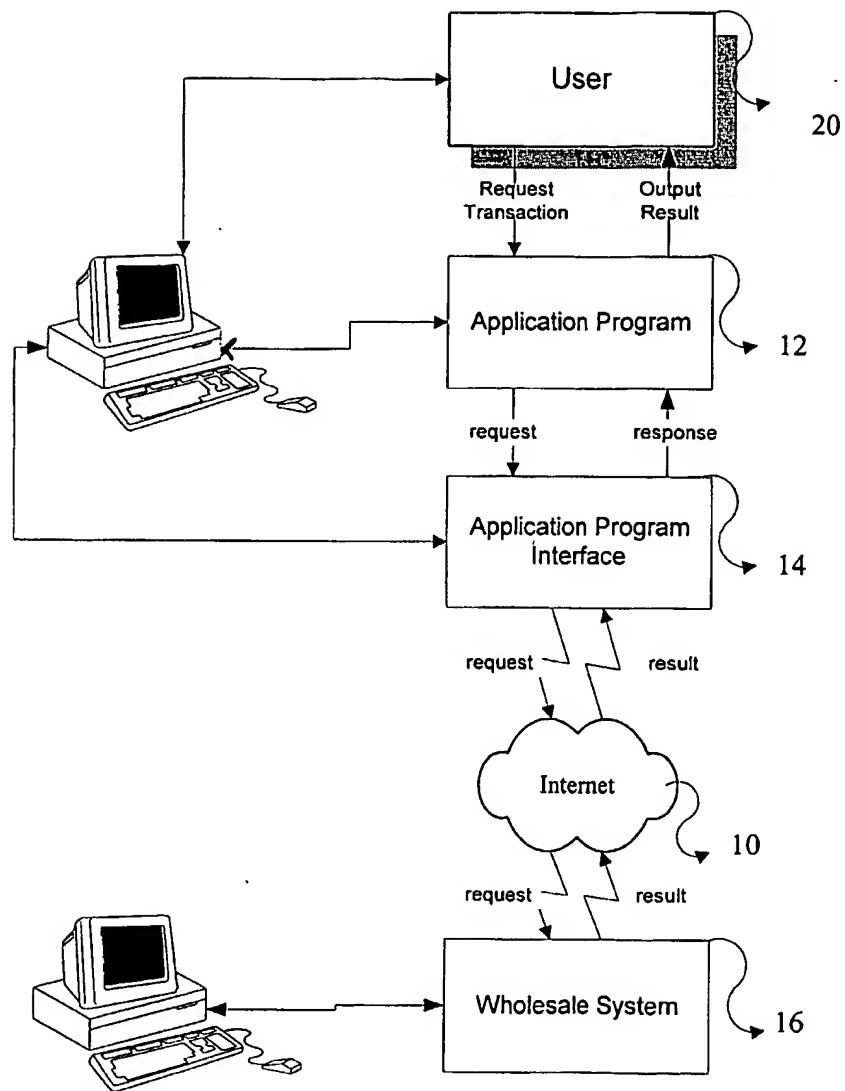
a component for initiating a transaction following such authentication and

following receipt of a request from the applications program; and

a component for transmitting commands for effectuating the transaction
via the secure socket layer to the system.

**FIG. 1**

**FIG. 2**

**FIG. 3**

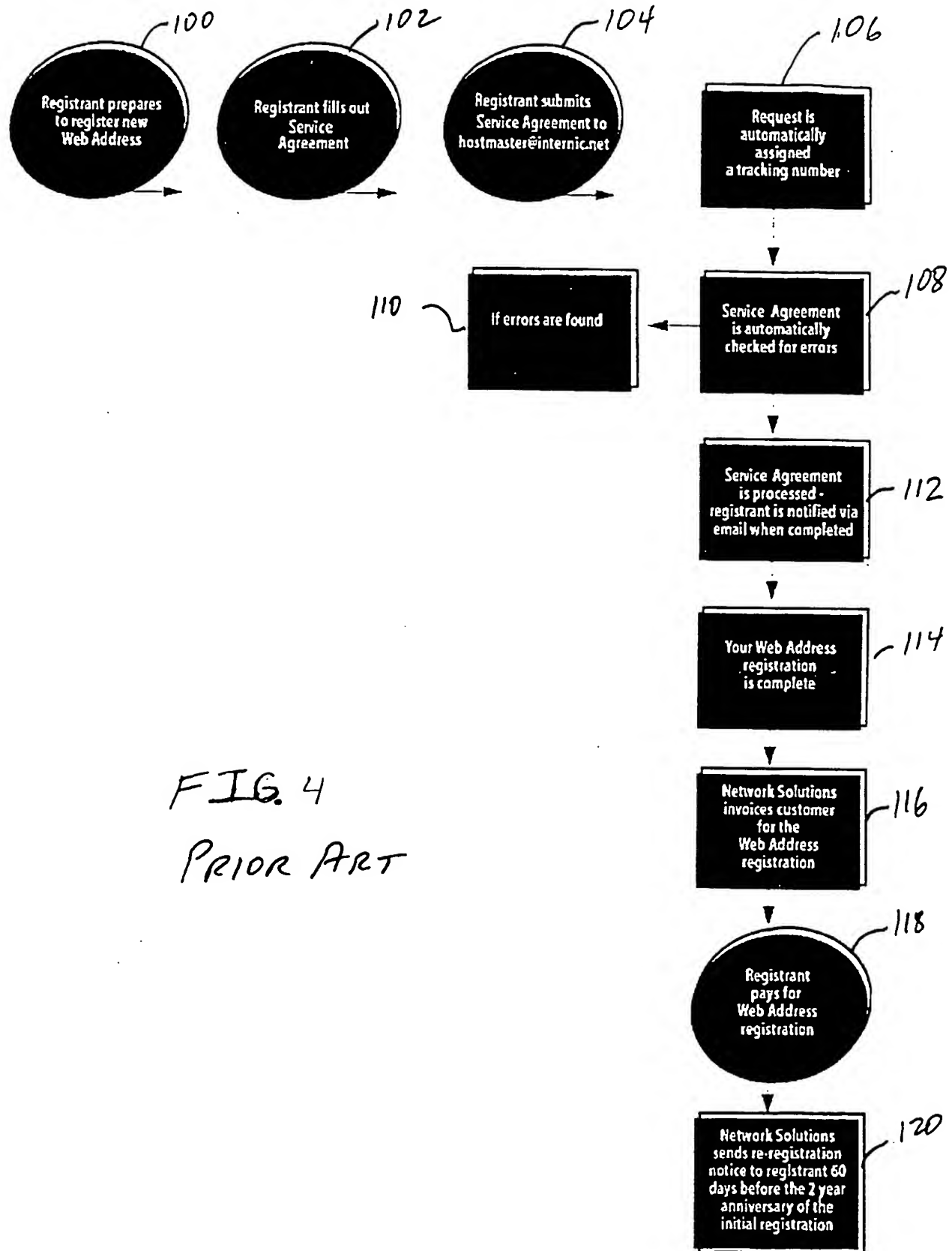


FIG. 4
PRIOR ART